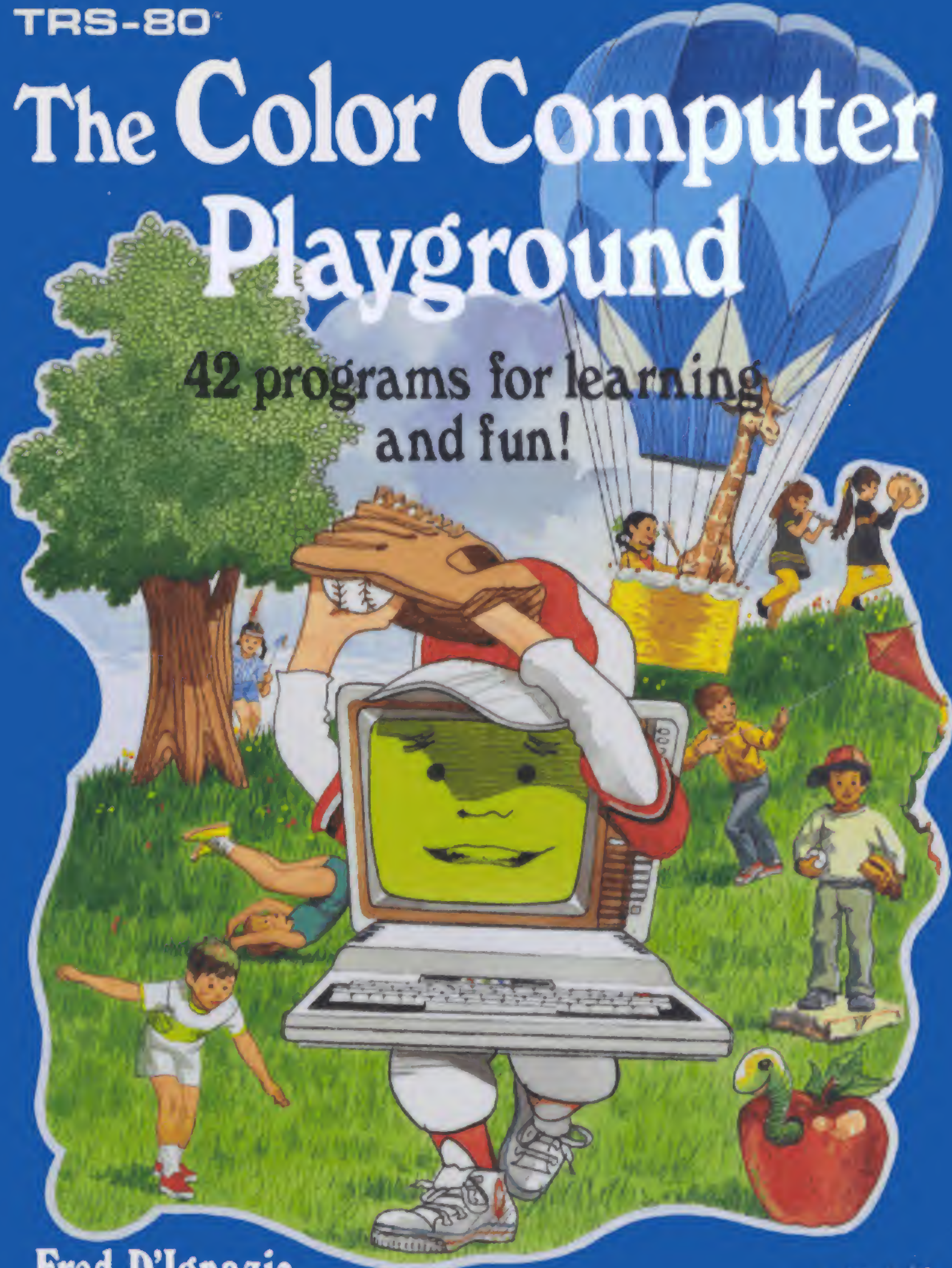
TRS-80®

The Color Computer Playground

42 programs for learning and fun!



Fred D'Ignazio

Cat. No. 26-3196

TRS-80®

The Color Computer Playground

TRS-80®

The Color Computer Playground

Fred D'Ignazio

Radio Shack®

For my team of young programmers, Howard Boggess, Angela Bradshaw, Joni Burdette, Brian François, Beth Ann Hostutler, Howard Levine, Mack McGhee, Melissa Perdue, and Scott Rainey, and their hardworking coach David James.

Conversion by Edwin Morales from *The Atari Playground* and *Atari in Wonderland* by Fred D'Ignazio, published by Hayden Book Company, 1983.

Printed in the United States of America

4 5 6 7 8 9 PRINTING
86 87 88 89 90 91 92 YEAR

PREFACE

This is a book of learning games you can type in on your Color Computer. You and your children can use this book to learn with the computer and about the computer. And while you are learning you are going to have fun.

If your children need to practice their spelling, have them play Scrambled Bees. If they are learning about fractions, give them some practice dividing up a pizza pie in Pepperoni, Please!

There are also programs on phonetics (Fat Cat), colors and sounds (The Pied Piper), multiplication (The Hamburger Contest), learning state names (Al's Tour of the States), Spanish (Uno! Dos! Tres!), and French (Un! Deux! Trois!).

There is even a child-size word processor program called Book Report. Children can use it to write poems, stories, and, of course, book reports.

There are forty-two learning games in this book. The games are grouped by subject area. They are educational, simple, and designed to engage your children's imaginations.

Most of the games are short. That means they are easy to enter into the computer. You can learn a lot about programming in BASIC by reading the description of each game and typing it in.

Each game has its own chapter. The chapter starts with a **For Parents and Teachers** section that briefly describes the game and the kinds of things children might learn by playing it.

Next is a **For Kids** section that weaves a story around each game and encourages children to use their imaginations when they play the game.

The Program comes next. It is the listing of the commands in the game. Following the program listing (in several chapters) is a **Typing Hints** section that explains how to type special parts of the program into the computer. Then comes a **Highlights** section that points out the major sections of the program. This is followed by a **Variables** section that lists all the variables (the pigeonholes in the computer's memory that store important numbers and letters). The chapter ends with a **Do-It-Yourself** section that gives you suggestions for expanding or enhancing the original game.

BASIC statements and commands in this book are written in capital letters: for example, FOR, NEXT, SAVE, and END. References to buttons on the keyboard are written in bold capital letters: for example, **SHIFT**, **BREAK**, and **ENTER**.

Many of the games use graphics (picture-making) characters or reverse video. The **Typing Hints** sections tell you what keys to type to obtain these characters.

Thanks to My Young Assistants ...

Young people designed several of the learning games in this book. My editor at Hayden, Gary Markman, suggested that I find some high school students to help me with the programming. I spoke to David James, the computer science instructor at Patrick Henry High School, which is only two blocks from my home, in Roanoke, Virginia. Within a few days David and his students were designing programs for this book.

The students came to school early and left late. They even got permission to leave other classes in order to work on the original programs for this book. David spent several weeks carrying their one computer back and forth between the school and the students' homes, so the students could work on the programs in the evenings and on weekends. At one point, all the programs mysteriously disappeared, probably due to a defective program recorder.

But, somehow, the programs all got done. To celebrate the completion of the programs and to thank the students, I took David and his team of young programmers out for a pizza dinner, compliments of Hayden.

According to David, this book is just the start for his students. Now they are anxious to begin working on new books and maybe even start their own "Learning Games" software company.

Make These Programs Your Own ...

These programs are intentionally short and simple. That makes them easy to type in, easy to understand, and easy to use. It also makes them easy to modify. As you are typing them in, make these programs your own. For example, add comment lines (REM statements) **in your own words** that explain what each program does. Also, at the beginning of each game, add PRINT commands to explain the game rules to your children.

I hope you and your children have as much fun using this book as I had writing it!

Fred D'Ignazio

EQUIPMENT NEEDED

To use the programs in this book, you will need the following equipment:

- A TRS-80 Color Computer or Color Computer 2
- A black and white or, preferably, a color monitor (TV)
- A cassette recorder or TRS-80 Color Computer diskette drive

CONTENTS

FACES . . . **1**

ALPHABET . . . **9**

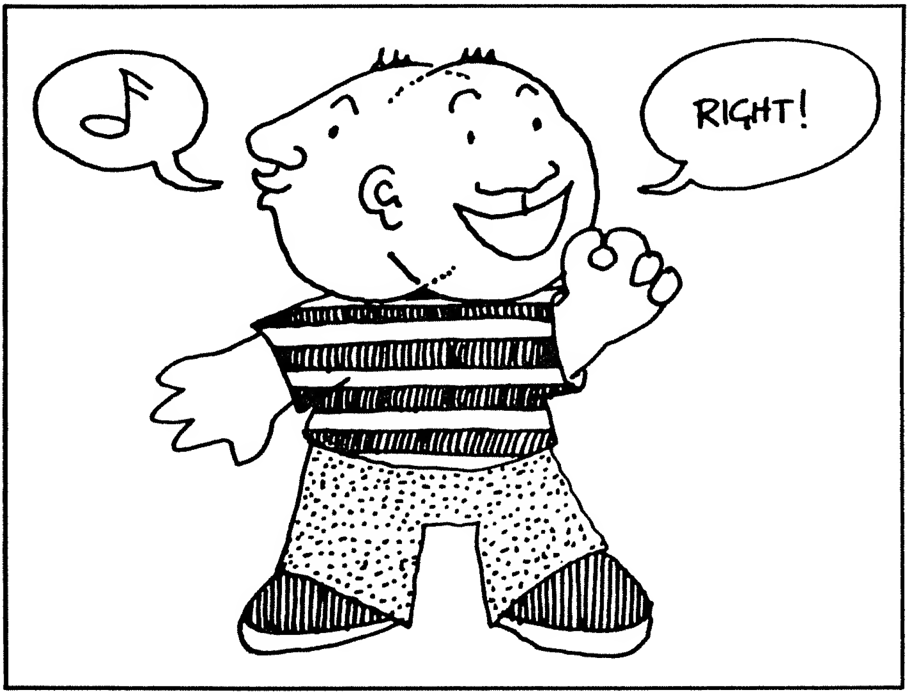
WORDS . . . **23**

NUMBERS . . . **57**

TRS-80®

The Color Computer Playground

FACES



1

HAPPY FACE

For Parents and Teachers ...

This is a game subroutine or "helper program." You can attach the subroutine onto most of the game programs in this book. When children get a correct answer, the subroutine prints a happy face, plays a happy tune, and congratulates the child.

For Kids ...

When you do something right, you expect a smile and a word of praise, right?

Here's a little program that draws a happy face on the TV screen, prints out the message "RIGHT!" and plays a happy "whistle."

The Program . . .

Program Name: **HAPPY**

```
1000 REM ***HAPPY FACE***
1010 FOR A=0 TO 2
1015 PRINT @A*32+365,CHR$(128);
1020 PRINT CHR$(128);CHR$(128);
1025 PRINT CHR$(128);
1030 NEXT A
1050 REM MOUTH RED
1055 SET(27,26,4):SET(28,27,4)
1060 SET(29,27,4):SET(30,27,4)
1065 SET(31,27,4):SET(32,26,4)
1070 REM NOSE ORANGE
1075 SET(29,25,8)
1110 SET(28,23,3)
1140 SET(31,23,3)
1200 PRINT@492,"RIGHT!";
1260 FOR W=180 TO 240 STEP 8
1270 SOUND W,1
1280 NEXT W
1290 FOR A=1 TO 900:NEXT A
1300 FOR A=1 TO 4:PRINT @(333+32*A)," ":
NEXT A:PRINT @492,"      ";
1350 RETURN
```

Typing Hints . . .

To make the happy face, you need to use graphics characters.

You can set a graphics point by using the SET(x,y,c) command, where x is the column, y is the row, and c is the color.

Highlights . . .

This is one of the simplest programs in the book. On lines 1010 to 1140, it prints out the happy face. On line 1200, it prints the message "RIGHT!" On lines 1260 to 1280 it makes the whistle sound.

The program uses the sound command—SOUND—to make the whistle.

Look at all the FOR and NEXT commands. These commands make the computer do something over and over. This is called a computer **loop**. The loop begins with the FOR command. It ends with the NEXT command. Every command between the FOR and NEXT commands is part of the loop.

The FOR command determines how many times the computer circles around the loop. For example, look at FOR A = 0 TO 2 on line 1010. This tells the computer to set the loop counter A to 0. Each time the computer races around the loop it adds one to the loop counter (A). The computer does three loops (from A = 0 to A = 2). Then it goes on and does something else.

Also, note that the program starts on line 1000 and ends on line 1350 with the command RETURN. That is because this program is really a subroutine—a helper program. It is not supposed to run on its own. It is supposed to help another program. (You can still test this subroutine by itself. When you finish typing it in, just type GOSUB 1010.)

You will want to attach this subroutine onto almost all the other programs in the book. That way, whenever you get the right answer in any of the programs, the computer will draw a happy face, whistle, and shout "RIGHT!"

Be sure to save this subroutine on a disk by using the SAVE"HAPPY" command, or by using the CSAVE"HAPPY" command if you are using a tape recorder.

You should continue to the next section and type in the Sad Face subroutine. Follow the directions in the Highlights section of Sad Face.

Variables ...

W	Counter for sound loop.
A	Counter for graphics loops—points to screen locations.

Do-It-Yourself ...

Try this happy face for awhile. When you get tired of it, change it to a new face. All you have to do is type in new SET commands. Change the eyes. Change the nose. Change the mouth. How about hair? Or a hat? And some ears?

2

SAD FACE

For Parents and Teachers ...

This game subroutine is to be used along with the Happy Face subroutine. It can be attached to most of the games in this book. When children answer the computer's question incorrectly, the subroutine draws a sad face on the TV screen, makes a sad sound, and prints the message "SORRY ... TRY AGAIN."

For Kids ...

When you miss a question or get the wrong answer, what do you expect? Probably a sad face. Maybe a sad sound. And, hopefully, encouragement to try again.

Here is a little program that does all that. It prints a sad face on the TV screen. Under the face it prints the message "SORRY ... TRY AGAIN." And it makes a sad sound.

The Program ...

Program Name: **SAD**

```
2000 REM ***SAD FACE***
2010 FOR A=0 TO 2
2020 PRINT @A*32+365,CHR$(128);
2025 PRINT CHR$(128);CHR$(128);
2030 PRINT CHR$(128);
2035 NEXT A
2050 REM MOUTH RED
2055 SET(28,26,4):SET(29,25,4)
2070 SET(30,25,4):SET(31,26,4)
2075 SET(27,27,4):SET(32,27,4)
2105 REM EYES BLUE
2110 SET(28,23,3)
2140 SET(31,23,3)
2150 PRINT@486,"SORRY ... TRY AGAIN";
2160 REM SAD SOUND
2170 SOUND 32,8
2180 SOUND 16,16
2220 FOR A=1 TO 500:NEXT A
2230 FOR A=1 TO 5
2235 PRINT@(288+32*A)," ":NEXT A:PRINT@4
86,"                   ";
2250 RETURN
```

Typing Hints ...

To make the sad face, you need to use graphics. The SET(x,y,c) command turns a small rectangle on the screen to the color selected by c, at screen positions determined by x and y.

Load in your HAPPY program first, then type in the SAD program.

Highlights ...

The sad face is similar to the happy face. Lines 2010 to 2140 print out the sad face. Line 2150 prints the sad message. Lines 2160 to 2220 play the sad sound. Lines 2230 and 2235 erase the sad face and the message.

To test the Sad Face, just type GOSUB 2010.

Note that the Sad Face starts on line 2000. The Sad Face is a subroutine, just like the Happy Face. You can attach the two to the end of other programs.

Save them with a SAVE command if they are on a disk, or with a CSAVE command if they are on tape, under the name "FACES".

Before you type another program, copy FACES back into memory with a LOAD command (disk) or a CLOAD command (tape).

Using the LOAD or CLOAD commands, you can add the happy and sad faces to other programs.

First, you LOAD "FACES" or CLOAD "FACES".

Then, you type in a new program.

Last, you save the combined, new program with a SAVE command (disk) or with CSAVE (tape).

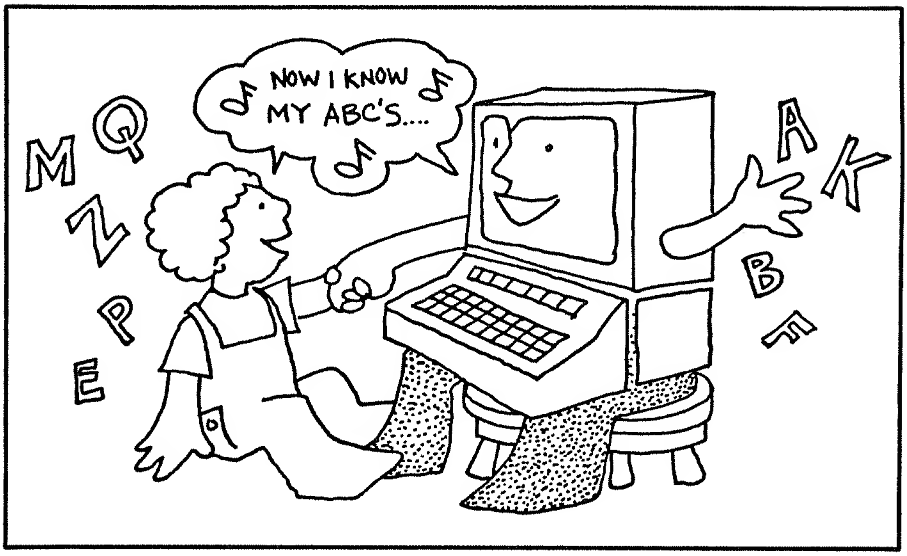
Variables ...

A	Counter for graphics loops—points to screen locations.

Do-It-Yourself ...

Try this sad face for awhile. When you get tired of it, change it to a new face. All you have to do is type in new SET commands. Change the eyes. Change the nose. Change the mouth. How about hair? Or a hat? And some ears?

THE ALPHABET



3

ALPHABET AL

For Parents and Teachers ...

This game helps children learn the letters of the alphabet. The letters appear on the screen, and the children have to find them on the computer keyboard.

For Kids ...

Remember Johnny Appleseed? Johnny went all over America planting seeds. The seeds sprouted and grew into apple trees with bright red, juicy apples.

But what would have happened if Johnny had planted "letter" seeds instead of apple seeds? Then letters would have sprouted all over America in place of apples.

In the program below we have created a new American folk hero, "Alphabet Al." Al is crazy about letters (the way

Johnny was crazy about apples). He journeys across America with a bag of letter seeds strung over his shoulder. Each time he reaches a new state, he races across it, planting seeds to make a new letter of the alphabet.

Type in and run the Alphabet Al program. You won't see Al himself, but you can follow his path by watching "alphabet" seeds sprout into letters. Listen to the noises the letters make when they blossom.

When Al is finished planting each letter, his trail of letters disappears. But before Al goes to a new state, he stops and asks you what letter he has just planted. He keeps asking the question until you get it right. Then, off he goes again, planting new letters.

The Game ...

Program Name: **AL**

```
40 REM *** ALPHABET AL
50 CLS
65 PRINT@35,"***  ALPHABET AL  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 FOR X=65 TO 90
100 CLS
120 FOR I=1 TO 25
130 PRINT@ RND(500), CHR$(X);
140 SOUND RND(200),2
150 NEXT I
152 FOR PAUSE=1 TO 200:NEXT PAUSE:CLS
153 PRINT@227,"WHAT LETTER DID AL PLANT"
;:INPUT A$
154 IF A$=CHR$(X) THEN GOSUB 1010:GOTO 1
60
155 GOSUB 2010:GOTO 100
160 NEXT X
170 CLS
180 END
```

Highlights ...

Al politely introduces himself (on line 65) before madly dashing off to plant "letter" seeds.

On line 100, a CLS command erases the screen each time Al enters a new state.

On line 130, a PRINT @ command traces Al's path of letters across the screen. On line 140, a SOUND command creates each letter's musical note as it appears.

The numbers 65 to 90 are the ASCII codes for the uppercase letters. The program uses a FOR-NEXT command (FOR X=65 TO 90 ... NEXT X) to create a loop that prints the 26 letters in the alphabet.

CHR$(X) selects each letter in the alphabet. For example, when X = 65, CHR$(65) prints A's, CHR$(66) prints B's, CHR$(90) prints Z's.

On line 154, you see the command GOSUB 1010. This calls the Happy Face subroutine.

On line 155, you find the command GOSUB 2010. This calls the Sad Face subroutine. Before you type the Alphabet Al program, remember to load in the merged Happy and Sad subroutines (FACES) with the LOAD command or the CLOAD command.

When each subroutine has finished its job, it reaches a RETURN command and bounces back to the command immediately following the GOSUB command.

Variables ...

A$	Your guess—which letter Al planted.
X	Letter of the alphabet (ASCII code).
I	Counter of FOR-NEXT loop (Al's path of 25 letters for each state).
PAUSE	Delay loop counter.

Do-It-Yourself . . .

Al runs through the alphabet from A to Z. Figure out a way to make him plant letters from Z to A (i.e., backwards).

Hint #1: Look at the FOR-NEXT loop on line 70. Hint #2: A FOR-NEXT loop can go backwards. In the FOR command, remember to include STEP -1. This makes the counter go backwards from the higher number to the lower number, one number at a time.

How would you make Al print letters at random?

Hint #1: You need to change lines 70 and 160.

Hint #2: To compute a number representing a random letter of the alphabet, use the command RND(26)+64. This gives you a random number between 65 and 90 (the ASCII codes representing the upper-case letters).

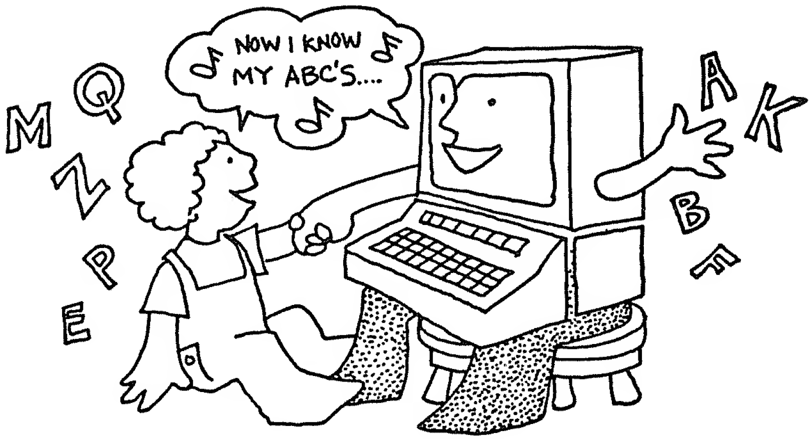
If Al's pace is too frantic and you'd like to slow him down a bit, you can add a delay loop at line 145 by typing between lines 140 and 150 this line:

```
145 FOR D=1 TO 200:NEXT D
```

You can also increase the amount of time the letters are on the screen before Al erases them and pops his question. Just increase the PAUSE loop on line 152 from 200 to 400 (or even higher).

4

THE ABC SONG



For Parents and Teachers ...

This game helps younger children learn the alphabet. It displays the letters of the alphabet while it plays the "Alphabet Song."

If your child isn't quick-tongued now, he or she will be after singing along with this program.

The program plays the ABC Song very slowly the first time. As it plays, it flashes the appropriate letter of the alphabet on the TV screen.

Your child will sing along and be proud that he or she can sing faster than the computer.

But, wait. The computer plays the ABC Song again. This time it plays faster.

Then it plays again—faster—and faster—and faster!

My young son has a ball trying to keep up with the computer. And he's becoming an alphabet expert, too!

The Game ...

Program Name: **ABCSONG**

```
50 REM *** THE ABC SONG
55 CLS
65 PRINT@37,"***  THE ABC SONG  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 EXTRA=5
90 CLS:RESTORE
100 FOR ALPHA=65 TO 90
110 PRINT@239,CHR$(ALPHA)
120 READ MUSIC:READ DURA
130 SOUND MUSIC, DURA*EXTRA
150 NEXT ALPHA
155 FOR D=1 TO 200:NEXT:PRINT@239," "
160 FOR LYRIC=1 TO 14
170 READ MUSIC:READ DURA:READ A$
175 PRINT@270,A$
180 SOUND MUSIC, DURA*EXTRA
195 PRINT@270," "
200 NEXT LYRIC
205 IF EXTRA <=.5 THEN 250
210 CLS:PRINT@269,"FASTER";:INPUT A$
220 IF LEFT$(A$,1) <> "Y" THEN 240
225 IF EXTRA>0 THEN EXTRA=EXTRA-.5:GOTO
90
235 GOTO 90
240 IF LEFT$(A$,1) <>"N" THEN 210
250 CLS:END
3000 REM *** ALPHABET NOTES
3010 DATA 89,4,89,4,147,4,147,4,159,4,15
9,4,147,8
3020 DATA 133,4,133,4,125,4,125,4,108,2,
108,2,108,2,108,2,89,8
3030 DATA 147,4,147,4,133,8,125,4,125,4,
108,8
3040 DATA 147,8,133,8,125,8,108,8
3060 DATA 89,4,NOW,89,4,I,147,4,KNOW,147
,4,MY,159,4,A,159,4,B,147,8,C'S
3070 DATA 133,4,NEXT,133,4,TIME,125,4,WO
N'T,125,4,YOU,108,4,SING,108,4,WITH,89,8
,ME
```

Highlights . . .

On lines 100 to 150 is a FOR-NEXT loop that reads the musical notes in from the DATA statements on lines 3010 to 3040. It prints the appropriate letter of the alphabet, and plays the note.

On lines 160 to 200 is a FOR-NEXT loop that reads the musical notes in from the DATA statements on lines 3060 to 3070. It prints the words "NOW I KNOW MY ABC'S, NEXT TIME WON'T YOU SING WITH ME"—and plays the notes.

When the computer finishes its song, on line 210 it asks you if you want to hear the song again, faster.

If you answer "YES" (or "Y"), the computer shortens the delay between the notes of the song by reducing the size of the variable EXTRA.

If you answer "NO" or if EXTRA is less than or equal to .5, then the computer ends the program. By the time EXTRA is reduced to .5, the ABC Song is sailing along, and only the most quick-tongued toddler (or adult) can keep up with it.

Take a look at lines 170, 220, and 240. You see that the variable A$ can be used to store the words in the song—"Now I know . . ."—or one-letter answers. LEFT$(A$,1) tells the computer to look at the characters starting at the first character and ending at the first character. This way you get the computer to look at only the first character in A$.

Variables . . .

EXTRA	Sets the speed of the ABC Song.
A$	Stores the words in the ABC Song ("Now I know . . .").
ALPHA	Counter—ASCII code for alphabet letters.
MUSIC	Stores musical notes (READ from DATA statements).
DURA	Stores duration of musical notes (READ from DATA statements).

LYRIC Counter—lets computer play the 14 words at the end of the song ("Now I know . . .").

D Counter—delay loop between letters and words of song.

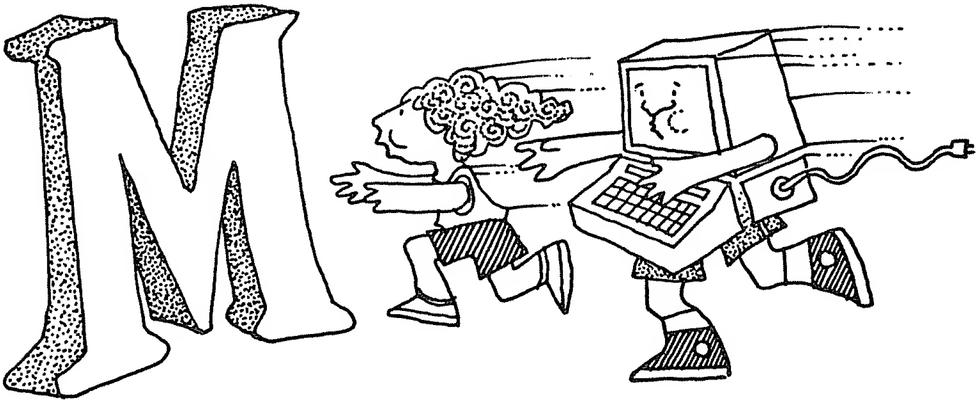
Do-It-Yourself . . .

The ABC Song runs through the alphabet from A to Z. Why not turn the song around and make it run through the letters backwards—from Z to A?

Also, if the song is too fast or too slow, just change the value of EXTRA. I've set it to 5 on line 70. This seemed to be the right starting speed for my child. But your children might like to sing faster.

5

FIND THE CAPITAL LETTER!



For Parents and Teachers ...

This game helps children learn the capital letters in the alphabet. It also teaches them the location of the letter keys on the computer keyboard.

For Kids ...

It's a race!

The race is between you and the computer. The computer flashes a letter on the TV screen. You have only a few seconds to find that letter on the keyboard and punch it in.

If you find the letter in time, the computer smiles, whistles, and says, "RIGHT!"

If you don't, the computer moans, looks sad, and tells you to try again. Then you get another chance.

If you are fast and you beat the computer, you can ask the computer to print the letters faster.

The Game ...

Program Name: FINDBIG

```
50 REM *** FIND THAT CAPITAL LETTER!
62 CLS
65 PRINT@34,"***  FIND THAT LETTER!  ***
"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 MAX=750
100 PRINT@236,"  "
120 FOR ALPHA=65 TO 90
125 COUNT=0
126 PRINT@239,CHR$(ALPHA)
128 COUNT=COUNT+1:IF COUNT=MAX THEN 160
130 CHILD$=INKEY$:IF CHILD$="" THEN 128
150 IF ASC(CHILD$)=ALPHA THEN GOSUB 1010
:GOTO 170
160 GOSUB 2010:PRINT@239," ":GOTO 125
170 NEXT ALPHA
171 IF MAX<=50 THEN 180
172 PRINT@236,"FASTER";:INPUT A$
173 IF LEFT$(A$,1)<>"Y" THEN 177
175 MAX=MAX-100:GOTO 100
177 IF LEFT$(A$,1)<>"N" THEN 172
180 CLS:END
```

Highlights ...

The key to beating the computer is speed. Your child needs to find a letter quickly and push the right button.

The Alphabet Al program also made the child enter a letter. But speed wasn't important, so the child needed to enter the letter *and* push the **ENTER** button.

This time we give the child a break. The child doesn't need to push the **ENTER** button, just the letter button.

We accomplish this with an INKEY$ command on line 130. Using INKEY$, the computer immediately grabs input from the keyboard without waiting for an **ENTER**. Each time you press a button, you send a character to the computer.

When you press the letters in the alphabet, you send a code for each letter to the computer. But, with the keyboard INKEY$, you are not sending the standard ASCII code, but the string value for that code. Line 150 takes the ASCII value of the variable CHILD$.

The computer prints the letters in the alphabet using a FOR-NEXT loop beginning on line 120. ALPHA, the loop counter, varies from 65 to 90. These are the ASCII codes for the 26 letters.

The computer prints the letter (line 126). Then, on line 128, it counts. If its count reaches 750 (the initial value of MAX), the computer jumps to line 160 and the sad face appears.

On line 130, the computer checks to see if a key has been pressed. If the value of CHILD$ is still the null character (""), the computer hops back to line 128 and continues counting.

If CHILD$ is not the null character, the computer (on line 150) checks to see if the child's answer ASC(CHILD$) matches the current letter of the alphabet. To do this, it has to take the ASCII value of CHILD$ and compare this number to ALPHA. For example, the ASCII code number for the letter A is 65; the counter ALPHA starts at 65. On line 126, the computer prints the letter A first, because ALPHA starts at 65.

On line 150, the computer comes to the command:

```
IF ASC(CHILD$) = ALPHA
```

If the child has pushed the A button, then 65 will be the ASCII value of CHILD$. And a 65 is also in ALPHA, so a match is made. The child has entered the right letter!

After the child has raced the computer through all the 26 letters in the alphabet, the computer (on line 172) asks if the child would like to race again. Only this time the race will be faster.

If the child says yes, the computer jumps back to line 100 and starts the race all over. But first it subtracts 100 from the highest number to which it counts after it prints each letter. The first time through the alphabet the computer counts to 750 after printing each letter. The second time through, it counts only to 650.

Remember to LOAD in the Happy and Sad subroutine (FACES) before typing in this program.

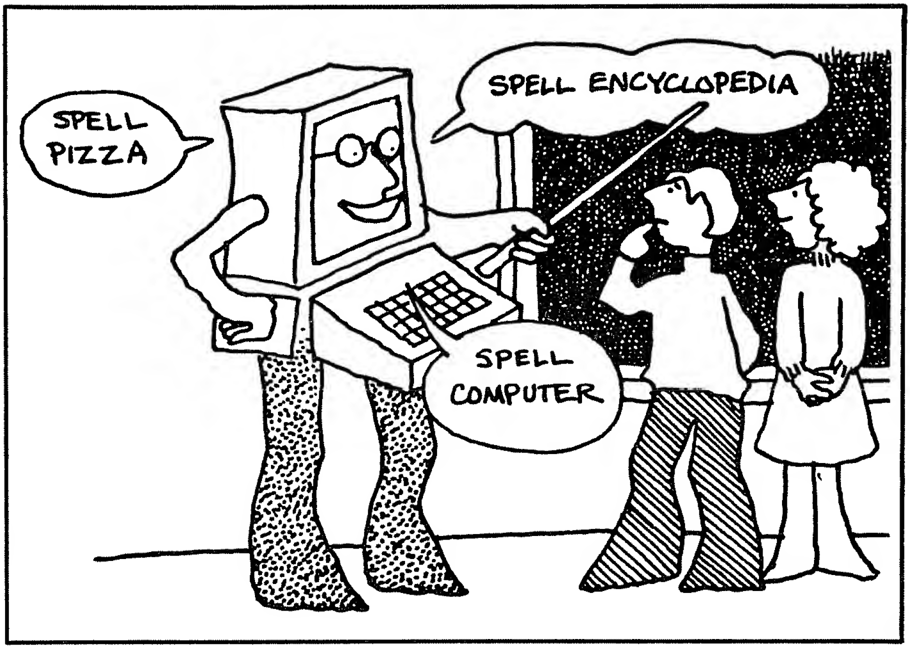
Variables ...

PAUSE	Delay loop counter.
MAX	How high the computer counts after printing each letter.
A$	Stores the child's yes (Y) or no (N) answer
ALPHA	Loop counter—varies from 65 to 90, the ASCII codes for the letters in the alphabet.
COUNT	Counter—for counting after printing each letter
CHILD$	The child's answer—stores the ASCII string code for the button the child pushed.

Do-It-Yourself ...

This game teaches a lot of things: letter recognition, the location of the letters on the keyboard, and hand-eye coordination. But after awhile the child might get tired going through the alphabet from A to Z. Why not jazz the program up by having it go through the alphabet backwards or at random? (Look at the Do-It-Yourself section in earlier chapters for programming hints.)

WORDS



6

THE SKYWRITER

For Parents and Teachers ...

This is an imagination game. The computer asks children for any sequence of letters or numbers. Then it displays the letters or numbers, over and over, all across the TV screen. This is a good game for teaching children how to spell their own names or the names of their pets and other members of their family.

For Kids ...

Have you ever seen a skywriter? A skywriter is an airplane that flies high up in the sky and leaves a white smokey trail. Simetimes the skywriter draws letters, words, and even pictures.

Wouldn't it be nice if you had your own skywriter and could draw your name in big, puffy cloud letters that everyone could see?

Buying a real skywriter would be expensive. But why not use your computer? The computer can draw your name on the screen really fast, just like a skywriter streaking across the sky.

But why stick just to your name? Let's have the computer "skywriter" draw any message you can think up.

The Game ...

Program Name: **SKYWRITR**

```
50 REM *** THE SKYWRITER
60 CLS
65 PRINT@36,"***  THE SKYWRITER  ***"
70 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 CLS
110 PRINT@353,"WHAT IS YOUR MESSAGE";:IN
PUT N$
115 IF N$=""THEN 180
120 CLS
130 FOR I=1 TO 150
150 PRINTN$;" ";
160 GO SUB 3000
172 NEXT I
174 FOR PAUSE=1 TO 2000:NEXT PAUSE
176 GOTO 100
180 END
3000 REM *** SKYWRITER SOUND
3020 SOUND89,1
3030 FOR K=1 TO 20:NEXT K
3040 RETURN
```

Highlights . . .

This is a simple program. It asks you for your message, and then it prints it on the TV screen 150 times. Each time it prints the message, it calls a subroutine (on line 3010) to play a musical tone.

When the program has finished printing your message, it pauses for a moment. Then it erases the screen and asks you for a new message. If you don't have a new message, just press the **ENTER** button and the program will end.

Variables . . .

PAUSE	FOR-NEXT delay loop counter.
N$	Stores your message.
I	FOR-NEXT loop counter—PRINT loop.
K	FOR-NEXT loop counter—sound loop in subroutine.

Do-It-Yourself . . .

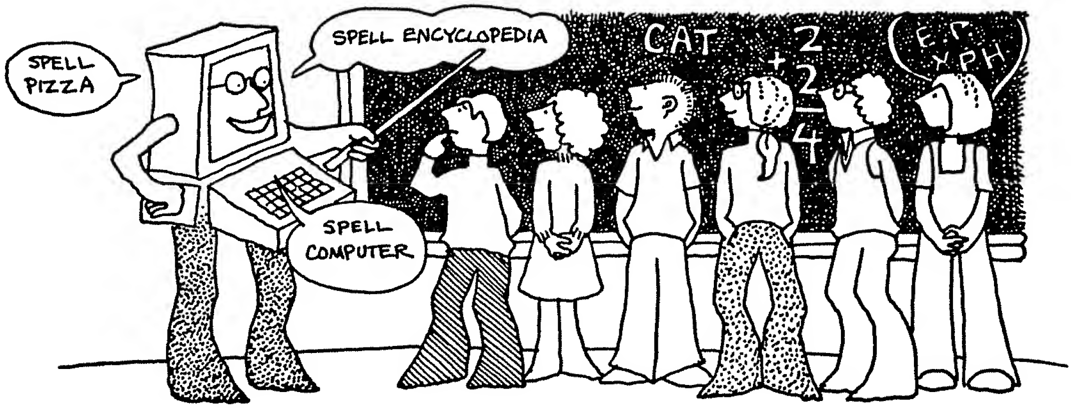
The airplane is invisible, but you can create an airplane out of graphics characters using SET statements. You can "fly" the airplane across the sky and print the child's message.

You can add other things to the sky: birds, clouds, and other planes.

The plane noise is pretty simple. You can experiment with the SOUND command to make the plane noise sound more like a real plane.

7

SPELLING BEE



For Parents and Teachers ...

This game helps children practice spelling. You or your children enter the words into the computer, and then the computer quizzes the children on each word. The children can take the computer quiz over and over until they have mastered all the words.

For Kids ...

The computer first asks you for 10 words you would like to know how to spell. Maybe they are the 10 words that will be appearing on your spelling test tomorrow morning.

You type in the 10 words, one at a time.

The computer lines you up against the wall and starts quizzing you. It shows you word number one—"PIZZA." Then the word disappears. The computer asks, “WORD?”

“PEETSAH,” you type.

The computer frowns, moans, and says, “SORRY ... TRY AGAIN.”

You try again: “PIZZA.”

“RIGHT!” says the computer. It smiles and whistles. Then it flashes word number two.

The Game ...

Program Name: **SPELLBEE**

```
50 REM *** SPELLING BEE
60 CLEAR 500
62 CLS
65 PRINT@37,"***  SPELLING BEE  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 FOR SPELL=1 TO 10
110 CLS:PRINT@322,"WORD #";SPELL;:INPUT
A$
120 ON SPELL GOSUB 3010,3020,3030,3040,3
050,3060,3070,3080,3090,3100
130 NEXT SPELL
140 FOR SPELL=1 TO 10
150 ON SPELL GOSUB 4010,4020,4030,4040,4
050,4060,4070,4080,4090,4100
160 CLS:PRINT@322,"WORD: ";A$
170 FOR PAUSE=1 TO 800:NEXT PAUSE
180 CLS:PRINT@322,"WORD";:INPUT B$
190 IF A$=B$ THEN GOSUB 1010:GOTO 210
200 GOSUB 2010:GOTO 160
210 NEXT SPELL
220 CLS:PRINT@418,"SAME WORDS AGAIN";:IN
PUT A$
230 IF LEFT$(A$,1)="Y" THEN 140
240 IF LEFT$(A$,1)<>"N" THEN 220
250 CLS:END
3000 REM *** STORE SPELLING WORDS
3010 W0$=A$:RETURN
3020 W1$=A$:RETURN
3030 W2$=A$:RETURN
```

```
3040 W3$=A$:RETURN
3050 W4$=A$:RETURN
3060 W5$=A$:RETURN
3070 W6$=A$:RETURN
3080 W7$=A$:RETURN
3090 W8$=A$:RETURN
3100 W9$=A$:RETURN
4010 A$=W0$:RETURN
4020 A$=W1$:RETURN
4030 A$=W2$:RETURN
4040 A$=W3$:RETURN
4050 A$=W4$:RETURN
4060 A$=W5$:RETURN
4070 A$=W6$:RETURN
4080 A$=W7$:RETURN
4090 A$=W8$:RETURN
4100 A$=W9$:RETURN
```

Highlights . . .

This program does everything in two small loops.

First, on lines 100 to 130, the computer lets you enter each word you want included in the Spelling Bee. It stores your answer (A$) in the "W" variables, from W0$ to W9$. It expects 10 spelling words.

The computer knows which W variable to store the word in using the FOR-NEXT loop counter, SPELL. Look at line 120. There are 10 line numbers to jump (GOSUB) to. SPELL acts like an index to show the computer where to jump. For example, if SPELL equals 1, then the computer jumps to the first line number (3010). If SPELL equals 2, the computer jumps to the second line number (3020). Finally, when SPELL equals 10, the computer jumps to the tenth line number (3100).

Each of the lines (3010 to 3100) is a one-line subroutine. Each subroutine copies the spelling word from A$ to the appropriate W variable. Then the computer reaches RETURN and bounces back to get more spelling words.

The computer uses the second loop (lines 140 to 210) to quiz you on the spelling words you just entered.

This time it uses the loop counter, SPELL, to transfer the right spelling word from a W variable back to A$.

On line 160, the computer flashes the spelling word on the screen. Then, on line 180, it erases the word and asks you to spell it.

If you are wrong, the computer jumps to the Sad Face subroutine. Then it gives you a chance to spell the word again.

After you have spelled all 10 words correctly, the computer asks you (on line 220) if you want to practice the words again. If you do, it jumps back to line 140 and begins the spelling bee a second time.

The computer will keep quizzing you until you drop. Or until you've mastered all the words.

Remember to LOAD the Happy and Sad subroutine (FACES). They fit between the main program (lines 50 to 250) and the W subroutines beginning on line 3000.

Variables . . .

A$	Stores correct spelling of spelling-bee words.
B$	Stores your answer.
W0$- W9$	Store the 10 spelling-bee words.
SPELL	Loop counter—index to subroutine for storing and retrieving spelling-bee words.
PAUSE	Delay loop counter.

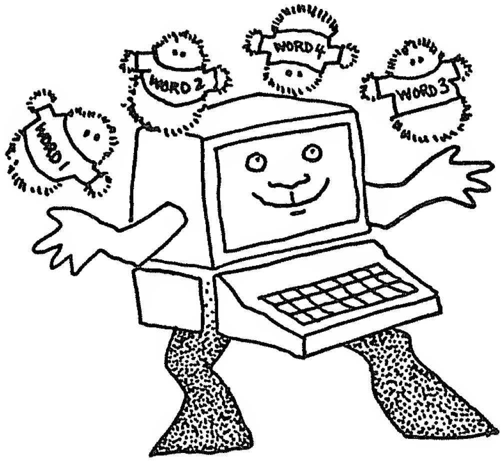
Do-It-Yourself . . .

There are lots of things you can do to this program. You can expand it so that it quizzes you on more than 10 words at a time. To expand it, you need to change the number 10 on lines 100 and 140 to a larger number. You also need to change the W variables to array variables, W$(0), W$(1), and so on. There's lots of room for adding new variables in this manner.

Another thing you can do would be to grade your child's score. You need to have a counter, COUNT, add one each time your child gets a word wrong and eliminate the GOTO 160 on line 200 so the child gets one chance at spelling the word correctly. After the test, you form a percentage by computing ((10–COUNT)/10)*100. Then you print out the percentage on the TV screen for your child to see.

8

SCRAMBLED BEES



For Parents and Teachers ...

This game helps children practice spelling. You or your children enter ten words into the computer. The computer arranges the words in a random order and then quizzes the children on each word. The children can take the computer quiz over and over until they have mastered all the words.

For Kids ...

In this chapter we create a "scrambled" spelling bee. The computer juggles the words around. Then it gives you the spelling bee with the words in the new order. And if you ask the computer to quiz you again on the same words, it will scramble the words again. Then it will give you the spelling bee.

The Game ...

Program Name: **SCRAMBEE**

```
50 REM *** SCRAMBLED SPELLING BEE
52 DIMA$(10)
55 CLS
62 PRINT@100,"***  SPELLING BEE  ***"
66 FORPAUSE=1TO1000:NEXTPAUSE
81 S$="975318642090817263543928170645675
8493021"
100 FORSPELL=1TO10
110 CLS:PRINT@260,"WORD # ";SPELL;:INPUT
 A$(SPELL)
130 NEXT SPELL
135 M=RND(4)-1
140 FORSPELL = 1TO 10
145 S1=VAL(MID$(S$,M*10+SPELL,1))+1
160 CLS:PRINT@260,"WORD:  ";A$(S1)
170 FOR PAUSE=1TO800:NEXT PAUSE
180 CLS:PRINT@260,"WORD ";:INPUTB$
190 IF A$(S1)=B$THENGOSUB1010:GOTO210
200 GOSUB2010:GOTO160
210 NEXT SPELL
220 CLS:PRINT@100,"SAME WORDS AGAIN";:IN
PUT A$
230 IF LEFT$(A$,1)= "Y" THEN 135
240 IF LEFT$(A$,1)<>"N" THEN 220
250 CLS:END
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Highlights ...

The program uses any array A$( ) to store the words. On line 52, the computer is told to set aside 10 spaces in the array A$( ) by a DIMension statement.

This program does everything in two small loops.

First, on lines 100 to 130, the computer lets you enter each word you want included in the spelling bee. It stores your answer (A$) in the "W" variables, from W$(1) to W$(10). It expects 10 spelling words.

The computer uses the second loop (lines 140 to 210) to quiz you on the spelling words you just entered and scramble the words to be spelled. It has four possible "canned" sequences to use. These sequences are stored in the variable S$ on line 81.

The computer chooses the sequence on line 135.

On line 145, the computer uses the following function to pick each of the 10 spelling-bee words:

```
S1= VAL(MID$(S$,M*10+SPELL,1))+1
```

We interpret this function (as the computer does) from the inside out. First, we know the variable M was chosen on line 135. It is a random number between 0 and 3. Let's say that M equals 3.

Second, let's say that this is the first time through the FOR-NEXT loop (lines 140 to 210). SPELL is the loop counter, so SPELL is equal to 1.

If we plug in the values of SPELL and M, we get the function:

```
S1= VAL(MID$(S$,3*10+1,1))+1
```

This translates to VAL (MID$(S$,31,1))+1. We know that (MID$(S$,31,1) selects the 31st character in the string S$. The 31st character in S$ is a 6.

If we plug in the 6 we get the function:VAL ("6")+1. The VAL function converts a number in string (nonarithmetic) form into a true number that can be used in arithmetic. The result of VAL ("6") is the number 6.

If we plug in the number 6 we get the function: 6+1. This, of course, translates to the number 7, so S1 = 7.

Now we are ready to pick our spelling-bee word. Look at line 160. The computer prints out A$(S1) which is the seventh word.

Voila!

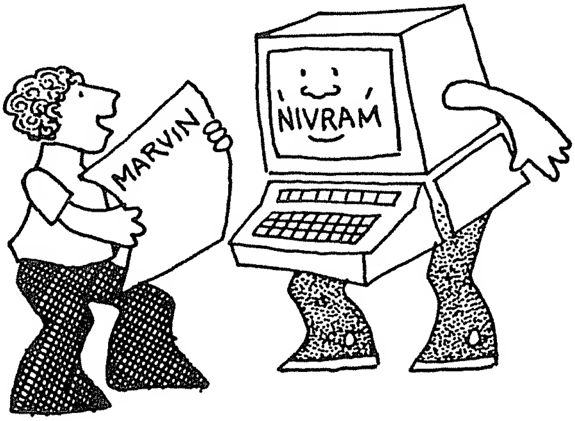
Note that on line 230, we now send the computer back to line 135. Line 135 is where the computer juggles the words and chooses a new scrambled word sequence.

Variables ...

A$( )	Array—stores correct spelling of words.
B$	Stores your answer.
M	Selects a random pattern for the words to be displayed.
SPELL	Loop counter—index for storing and retrieving spelling-bee words.
PAUSE	Delay loop counter.
S1	Subscript of the particular word picked from array A$( ).
S$	Stores the four possible sequences in which words can be displayed.

9

BACKWORDS



For Parents and Teachers ...

This game is mostly for fun. But it teaches children to look at words in a playful, original way. It shows them that words aren't fixed and frozen. They can be flipped over and turned around. The result is amusing and often surprising.

For Kids ...

Did you ever wonder what your name looks like spelled backwards? I did. So I tried this computer game.

When the computer typed "FRONTWORDS?" I typed my name—FRED.

A message flashed on the TV screen:

```
BACKWORDS MACHINE NOW WORKING
```

Moments later, the computer printed my name forwards and backwards. Forwards it was FRED. Backwards it was DERF!

What does **your** name look like backwards? Or your mom's name? Or your dad's? Or your favorite hero's name?

Use this game to find out.

The Game ...

Program Name: **BACKWORD**

```
50 REM *** THE BACKWORDS MACHINE
60 CLS
65 PRINT@225,"*** THE BACKWORDS MACHINE
***"
66 FORPAUSE=1TO1000:NEXTPAUSE
100 CLS
105 C=0:M$="":M2$=""
110 PRINT@390,"FRONTWORDS";:INPUTM$
160 CLS
170 PRINT@384,"backwords machine now wor
king"
180 FOR PAUSE=1TO1500:NEXT PAUSE
190 FOR I=LEN(M$) TO 1 STEP-1
193 C=C+1
195 IF MID$(M$,I,1)=" " THEN M2$=M2$+" "
:GOTO 210
200 M2$=M2$+MID$(M$,I,1)
210 NEXT I
220 CLS
230 PRINT@192,"FRONTWORDS:"
240 PRINT
250 PRINT"  ";M$
260 PRINT:PRINT
270 PRINT"BACKWORDS:"
280 PRINT
290 PRINT"  ";M2$
300 PRINT:PRINT
310 PRINT"PLAY AGAIN";:INPUT A$
320 IFLEFT$(A$,1)="Y" THEN100
330 IFLEFT$(A$,1)<>"N" THEN 310
340 CLS:END
```

Typing Hints . . .

In line 170, the print statement "BACKWORDS MACHINE NOW WORKING" is to be typed in **reverse video**, although it appears in lower-case letters in the listing.

To enter letters or symbols in reverse video, you need to press the **SHIFT** and **0** keys simultaneously, type in the letters of the message itself, and then press the **SHIFT** and **0** keys again to restore the keyboard to normal video.

Highlights . . .

The key to this program is the loop on lines 190 to 210. The name you typed in is stored in M$. The computer takes the last letter of the name in M$ and places a copy of it in the **first** position in M2$. Then it takes the next-to-the-last letter in M$, copies it, and then places it in the **second** position in M2$. The computer continues this process until it has reached the very first letter in M$. It puts this at the **end** of the letters it has stored in M2$. In this way, the computer turns the entire word around. What went frontwards into M$ comes out backwards into M2$.

Do-It-Yourself . . .

You might modify this program so that you can store the "backwords" on tape or disk.

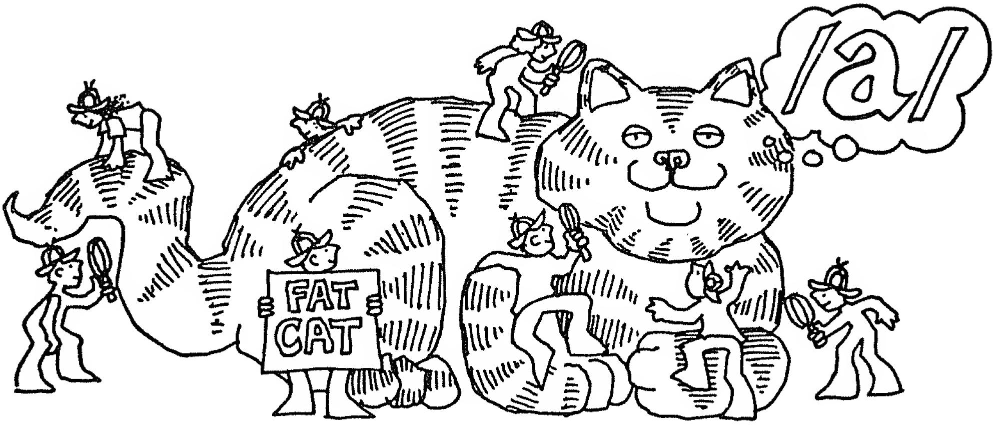
Also, "backwords" are just an example of the limitless ways you can play around with words on your computer. Electronic words aren't fixed or frozen. They are plastic or clay—infinitely malleable.

One way to get your computer to play with words is to translate all the words from regular video into **reverse video**

For example, the upper-case letters in the alphabet (A to Z) are represented by ASCII values of 65 to 90. The upper-case letters in reverse video are represented by 97 to 122. You can change the Backwords program so that every time you type in a letter A, for example, the program changes an "A" to an "A." The same goes for all the other letters. This way, children can get their names typed out in reverse video.

10

FAT CAT



For Parents and Teachers . . .

This game helps children learn phonetic symbols and phonetic sounds. The computer displays the phonetic symbols on the TV screen. Then it flashes a word on the screen. The child says the word aloud and tries to guess which phonetic sound is hidden inside the word. An arrow points to the first phonetic symbol. To move the arrow to a new symbol, the child presses the **SPACE** bar. To select a symbol the child presses the **ENTER** button.

For Kids . . .

Did you ever notice how certain words rhyme or sound alike? Words like **red** and **bed**, or **big** and **pig**, or **fat** and **cat**?

The reason the words rhyme is that they have the same phonetic sound (or **phoneme**) hidden inside them. For example, hiding inside the words red and bed is the phoneme /**e**/. Hidden inside the words big and pig is the phoneme /**i**/. And inside the words fat and cat is the phoneme /**a**/.

In this game you get to play detective. The computer will flash a word on your TV screen. Say the word out loud a couple of times. Now look at the list of phonemes on the screen. Your job as a sound detective is to discover which phoneme is hidden inside the word.

The Game ...

Program Name: **FATCAT**

```
50 REM *** FATCAT GAME
60 CLS
65 PRINT@36,"*** THE FAT CAT GAME ***"
66 FORPAUSE=1TO1000:NEXTPAUSE
68 FORJ=1TO8:FORK=1TO5:READW$(J,K):NEXTK
:NEXTJ
100 FORM=1TO5
102 TRY=0
110 GOSUB 3010:REM * SET UP SCREEN
112 X=64:REM * LINE - 1
115 IFTRY=0THENGOSUB 4010:REM * SELECT W
ORD
120 GOSUB 4110:REM * PRINT WORD
125 GOSUB 3510:REM * MOVE ARROW
130 IF J=C THEN GOSUB1010:GOTO 150
140 GOSUB 2010:TRY=1:GOTO112
150 NEXT M
160 PRINT@448,"PLAY AGAIN <Y/N>";:INPUT
A$
170 IFA$="Y"THEN PRINT@448,"       ":GOT
O 100
180 IFA$<>"N" THEN 160
190 CLS:END
1010 RETURN
2010 RETURN
3000 REM *** SET UP PHONETIC SOUNDS
```

```
3010 PRINT@100,"/a/"
3020 PRINT@132,"/ch/"
3030 PRINT@164,"/e/"
3040 PRINT@196,"/f/"
3050 PRINT@228,"/l/"
3060 PRINT@260,"/sh/"
3070 PRINT@292,"/s/"
3080 PRINT@324,"/th/"
3090 PRINT
3110 PRINT"y=yes    n=no"
3120 RETURN
3500 REM *** ARROW SUBROUTINE
3510 P$="->":B$="  "
3511 C=0
3512 OX=X
3520 X=X+32:IFX>320THEN X=96:C=0
3552 C=C+1
3560 PRINT@OX+2,B$;
3562 PRINT@X+2,P$;
3570 I$=INKEY$: IF I$=""THEN3570
3580 IF I$="Y" THEN PRINT@X+2,B$;:RETURN
3590 IF I$="N" THEN 3512
3600 GOTO 3570
4000 REM *** SELECT SOUND/WORD
4010 J=RND(8):REM PHONEME TYPE
4020 K=RND(5):REM WORD SELECT
4090 RETURN
4100 REM *** PRINT WORD
4110 PRINT@452,"            "
4120 PRINT@452,W$(J,K)
4130 RETURN
5010 DATA FAT,CAT,MAN,HAT,FAST
5020 DATA RICH,WITCH,LUNCH,CATCH,MUCH
5030 DATA RED,HEN,BED,LEG,PET
5040 DATA STUFF,OFF,CUFF,MUFF,HUFF
5050 DATA BELL,STILL,WILL,ROLL,SELL
5060 DATA SHIP,FISH,SPLASH,WISH,DISH
5070 DATA GRASS,GLASS,CROSS,DRESS,MESS
5080 DATA THIN,BATH,CLOTH,WITH,THICK
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

The message on line 3110 and the phonetic sounds on lines 3010 to 3080 appear in lower-case letters in the listing. This means they are to be typed in reverse video. Push the **SHIFT** and **0** keys simultaneously, type the letters, and then press the **SHIFT** and **0** keys again. This gets you back to normal video.

Highlights ...

Line 68 reads the list of 40 words into a two-dimensional array, W$.

This game uses four major subroutines. The first subroutine (called on line 110) prints the list of phonetic symbols. The second subroutine (called on line 115) selects the word with the mystery phoneme hidden inside. The third subroutine (called on line 120) prints the word on the TV screen. The fourth subroutine (called on line 125) moves the "choice" arrow each time you press the **SPACE** bar.

Variables ...

PAUSE	Delay loop counter.
A$	Child's yes-no answer to "PLAY AGAIN?"
W$	Array—holds all the words.
M	Loop counter—controls number of words/game.
TRY	If TRY=1, child has given an incorrect answer.
J, K	Pointers that indicate one word containing a certain phoneme.
C	Pointer to phoneme symbol (child's answer).
X	Current column position of arrow.
OX	Old position of arrow.

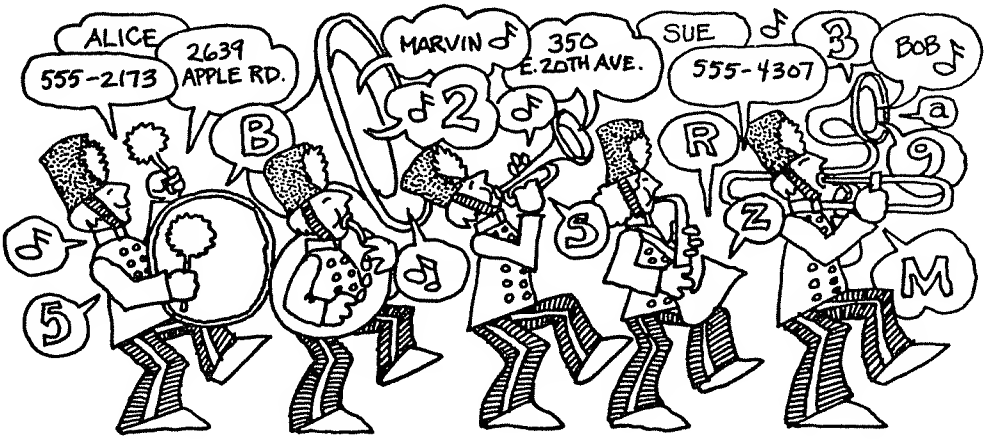
Do-It-Yourself . . .

This is only a small sample of the different phonetic symbols that represent word sounds. You can add new sounds by changing the list of sounds in the subroutine beginning on line 3000. You can add new words with these sounds by changing the DATA commands beginning on line 5010.

Also, please note that some of the words in the current list have more than one phonetic sound hidden inside. Yet the current program only recognizes one of those sounds as a correct answer. How would you modify the program to make it recognize the other sound or sounds?

11

MUSICAL WORDS



For Parents and Teachers ...

This game helps children learn important words and numbers, such as their name, phone number, address, etc. The child types in a word (or number) and the computer sets it to music and plays it like a song. The computer accepts only the correct letters or numbers. All other keys on the computer keyboard are "asleep."

For Kids ...

Pretend that you have a magic horn. When you blow it, out pops a pretty sound. But something else comes out, too: a letter.

You try playing the horn again. A new sound comes out. And out flies a different letter.

You play lots of different sounds on your horn. Letters come zooming out of the horn. They bump together and form a word. The word looks familiar It's your name. And it's musical!

You can make your name musical. But that's not all. Try setting your mom's name to music. Or your dad's name. Or your little sister's name. Or the name of your favorite pet gerbil. Next, try out your phone number, your street name, your city, and your state. Like your name, they can all be musical.

The Game ...

Program Name: **WRDMUSIC**

```
50 REM *** WORD MUSIC
60 CLS
65 PRINT@37,"***   WORD MUSIC   ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM MUSIC(15)
82 FOR I=1 TO15:READ MUSIC(I):NEXT I
100 CLS
110 PRINT@386,"YOUR WORD";:INPUT WD$
115 IF WD$="" THEN CLS:END
120 IF LEN(WD$)>16 THEN WD$=LEFT$(WD$,15
)
130 CLS
140 PRINT@386,"*** TYPE YOUR WORD ***"
145 PRINT@449," ";
150 FOR I=1 TO LEN(WD$)
160 K$=INKEY$:IF K$=""THEN 160
162 K=ASC(K$)
165 IF K=13 THEN 100
170 IF K$<> MID$(WD$,I,1) THEN 160
180 PRINTK$;
181 SOUND MUSIC(I),8
190 NEXT I
200 FOR PAUSE =1 TO 300:NEXT PAUSE
210 GOSUB 3510
220 GOTO 130
3500 REM *** PLAY WORD/NOTES
3510 CLS
```

```
3515 PRINT@34,"*** A LITTLE WORD MUSIC *
**"
3517 FOR J=1 TO 5
3518 PRINT@103," ":
3519 PRINT@103," ";
3520 FOR I=1 TO LEN(WD$)
3530 PRINTMID$(WD$,I,1);
3540 SOUND MUSIC(I),8
3560 NEXT I
3570 NEXT J
3580 RETURN
4000 DATA 89,108,147,176,193,204,218,227
,232,133,159,176,197,210,216
```

Highlights ...

The program begins by reading musical notes into the array MUSIC on line 82.

Next the program asks you for a word. If you are finished with the game, you just press the **ENTER** button.

The computer asks you to type in your musical word. You type in the same word as before, only this time each letter is set to music. The computer accepts only the letters in the word.

When you have entered the entire word, the computer prints the word five times, playing it like a song. Then it erases the word and asks you to enter it again. If you are tired of this word, press the **ENTER** button.

Variables ...

PAUSE Delay loop counter.

WD$ Stores word or number that you want set to music.

MUSIC Array—saves 15 musical tones to match with letters or numbers.

I In line 82 it is a counter for a READ loop; in line 150 it is a counter for the loop in which the child types in the letters or numbers; in line 3520 it is a counter for the loop that prints the letters or numbers and plays the musical notes.

K$ ASCII byte—key typed by child.

J Counter for loop that plays word five times.

Do-It-Yourself ...

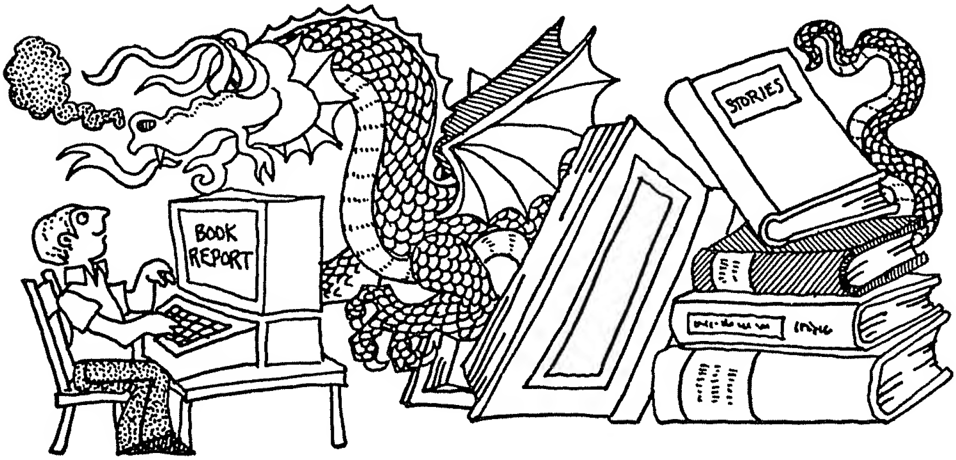
The program accepts a maximum of 15 characters (letters or numbers) at a time. You can increase the size of WD$ and MUSIC so that the computer can accept longer words and numbers. Then you need to change the PRINT command on lines 145, 3518, and 3519 so that, if the word or number can't fit on the current line, the computer continues it neatly on the next line (or skips a line).

Also, the program always uses the same 15 musical notes, no matter what letters or numbers the child enters. You can replace the DATA command on lines 4000 and 4010 with a longer list of musical notes. To find the note values, see the Appendix at the end of the book.

Another way of obtaining different notes for the letters is to use a random selection of the notes in the MUSIC array. Change line 3540 to use SOUND MUSIC(RND(15)),8.

12

BOOK REPORT



For Parents and Teachers ...

This game will help children with their writing assignments, especially book reports. The program asks the child for pertinent information—the date, the child's name, the author, and the title of the book. Then the computer asks the child "WHAT HAPPENED FIRST?" When the child enters this information, the computer asks "WHAT HAPPENED NEXT?"

The child responds to the computer's questions and enters the report, sentence by sentence. When finished, the child types END or THE END. Then the computer prints the entire book report, all nicely formatted, on the TV screen.

For Kids ...

Did you ever have to write a book report or tell a story, and you didn't know where to start?

Here's a sure-fire way to get you going. Pretend that the computer is a person. And imagine that the person can hardly wait to hear about a neat book you just read or a story you made up inside your head.

What does the person ask you to get you started? He says, "What happened first?" So you tell him.

It sounds exciting, so he asks, "What happened next?" So you tell him that, too.

Each time you tell him a little bit of the story, he wants to hear more. So he keeps asking "What happened next?"

Before you know it, you've finished describing the book you read or the story you made up. When he asks you "What happened next?" you simply say "END" or "THE END."

Then, sit back. The computer hasn't forgotten a word you said. It is going to replay the entire story or book report right before your eyes.

As it types the story on the TV screen, you can be busy copying it in your notebook. Then it's ready to turn in to your teacher at school tomorrow morning.

The Game ...

Program Name: **BKREPORT**

```
50 REM *** THE BOOK REPORT
55 CLEAR 4000
60 CLS
65 PRINT@35,"***  THE BOOK REPORT  ***"
67 FOR PAUSE = 1 TO 1000: NEXT PAUSE
70 DIM R$(20):REM 20 'WHAT HAPPENED NEXT
' ANSWERS MAXIMUM.
92 B$="  "
94 DIM PAR(20)
95 P=1:S=1
100 CLS
110 PRINT"WHO'S OUT THERE";:INPUT NM$
120 CLS
125 PRINT"TODAY'S DATE";:INPUT DT$
127 CLS
130 PRINT"BOOK TITLE";:INPUT T$
```

```
140 CLS
150 PRINT"AUTHOR";:INPUT A$
160 CLS
170 PRINT"WHAT HAPPENED FIRST":INPUT L$
190 R$(S)="    "+L$:PAR(S)=LEN(R$(S))
200 CLS
202 S=S+1
210 PRINT"WHAT HAPPENED NEXT":INPUT L$
212 R$(S)=L$
215 IF L$="END" OR L$="THE END" THEN 300
217 IF L$=""THEN P= P + 1:R$(S)="@":GOTO
 230
220 LS=LEN(L$):IF MID$(L$,LS,1)="."THEN
R$(S)=R$(S)+" "
230 PAR(S)=LEN(R$(S))
240 GOTO 200
300 CLS
303 L$="BOOK REPORT BY "
304 L$=L$+NM$
305 GOSUB 3010:REM * CENTER LINE
306 PRINT@X,L$
308 L$=DT$:GOSUB3010:PRINT@X,L$
310 L$=T$:GOSUB 3010:PRINT@X,L$
320 GOSUB3510:REM *** TITLE LINE
322 GOSUB3010
330 PRINT@X,L$
340 L$="BY ":L$=L$+A$:GOSUB3010:PRINT@X,
L$
345 PRINT:PRINT
350 GOSUB 4010:REM *** PRINT REPORT
355 IF PEEK(136)=5 AND PEEK(137)>192 THE
N GOSUB 4510
360 L$="THE END":GOSUB3010:PRINT@X,L$
390 GO TO 390
578 P=P+1:D=0:RS=0
3000 REM *** CENTER LINE SUBROUTINE
3010 X=PEEK(136)-4:X=X*256+16+PEEK(137):
TX=LEN(L$):X=X-INT(TX)/2
3020 RETURN
3500 REM *** TITLE LINE SUBROUTINE
3510 L$="":FORI=1TOLEN(T$)
3520 L$=L$+"-"
3530 NEXTI
3540 RETURN
```

```
4000 REM *** PRINT REPORT
4010 D=0:P=1
4012 IFR$(P)="END" OR R$(P)="THE END" TH
EN RETURN
4015 IF LEN(R$(P))<32 THEN GOSUB 5010:GO
TO 4012
4016 RS=31
4020 IFMID$((R$(P)),RS,1)=" "THEN4090
4030 RS=RS-1:GOTO 4020
4042 IF LEN(R$(P))>25 THEN PRINT:GOTO 50
78
4090 IF PEEK(136)=5 AND PEEK(137)>192 TH
EN GOSUB 4510
4095 IF PAR(P)>D+RS+3 OR PAR(P)=0THEN410
0
4096 GOSUB5010:GOTO4012
4100 PRINT MID$((R$(P)),D+1,D+RS)
4110 D=D+RS+1
4120 R$(P)=MID$((R$(P)),D):D=0
4122 IF LEN(R$(P))<32 THEN GOSUB 5010
4130 GOTO 4012
4500 REM *** PAUSE/CLEAR SCREEN
4510 FOR PAUSE =1 TO 3000:NEXT PAUSE
4512 PRINT@160,B$
4520 FORL=1TO9
4530 PRINTB$
4540 NEXTL
4542 POKE136,4:POKE137,192
4560 RETURN
5000 REM *** PARAGRAPH
5010 IF PEEK(136)=5 AND PEEK(137) >192 T
HEN GOSUB 4510
5012 IF R$(P) <> "@" THEN 5040
5014 IF R$(P+1)="END" OR R$(P+1)="THE EN
D" THEN 5074
5020 L$=R$(P+1):R$(P+1)="   "+L$
5022 PAR(P)=LEN(R$(P+1))
5024 GOTO 5074
5040 PRINTR$(P);
5044 IF R$(P+1)="@" THEN 5074
5046 IF R$(P+1)="END" OR R$(P+1)="THE EN
D" THEN 5074
5048 D=31-LEN(R$(P)):RS=1
5051 L$=" "
```

```
5052 IF MID$((R$(P+1)),RS,1)=" "THEN 505
8
5054 L$=L$+MID$((R$(P+1)),RS,1)
5056 RS=RS+1:IF RS<D THEN 5052 ELSE 5074
5058 PRINTL$;
5060 L$=R$(P+1):S=LEN(L$)-RS
5062 R$(P+1)=RIGHT$(L$,S)
5064 D=D-1-RS:IF D>4 THEN RS=1:GOTO 5051
5074 PRINT
5078 P=P+1:D=0:RS=0
5080 RETURN
```

Highlights ...

The computer copies the entire story or book report into a string array called R$, which has been DIMensioned in line 70 for 20 messages.

When you are done entering a book report or story into the computer and you type THE END (or, simply, END), the computer prints out a title, the date, your name, and the entire text.

The subroutine beginning on line 3000 centers all the headings, including the book title, the author, the date, and your name.

The subroutine beginning on line 3500 underlines the book title. The subroutine beginning on line 4000 prints the text of the report or story.

If a word is too long to fit at the end of a line, the computer doesn't break the word in two. Instead, it lifts the entire word and places it on the next line.

When your child is writing the report or story and comes to the end of a paragraph, he or she should answer the computer's question ("WHAT HAPPENED NEXT?") by pressing the **ENTER** button (see line 217).

When you press the **ENTER** button without entering any other information, the computer receives a null character and inserts an @ sign. When the computer prints out the report it knows where the @ signs are because it stored their locations in the string array R$. When it comes to an @ sign, it ends an old paragraph and creates a new, indented paragraph automatically.

If the book report or story is too big to fit on the screen, the computer pauses at the bottom of the screen for you to read what is there. Then it erases only the text on the screen (leaving the title information and your name intact), and fills the screen with new text. It keeps doing this until it comes to the end of the story or report. At the end, it prints the words "THE END," centered on the screen.

Did you notice how many PEEKs and POKEs there are to memory positions 136 and 137? The computer goes to locations 136 and 137 to find where it is on the TV screen as it prints out lines of text. The computer has only 16 lines. The book report uses locations 136 and 137 to keep the report or story from spilling onto a new screen.

Variables ...

PAUSE	Delay loop counter.
NM$	Child's name.
T$	Book title.
A$	Author's name.
R$	Array—the text of the story or report.
L$	A single text line.
DT$	The current date.
B$	A blank line (to erase a single screen line).
P	Points to the next sentence.
X	The position of centered text.
I	Loop counter for centering title line.
L	Loop counter in loop that erases old text from screen.

Q Holds the total number of text lines in the book report.

D Starting position of current line in string array R$.

RS Ending position of current line in text string array R$.

PAR Array—length of each string in the R$ array.

Do-It-Yourself ...

You have the rough beginnings of a word processor here. It lets you enter reports, letters, stories, or whatever you choose in an orderly, neat-looking format.

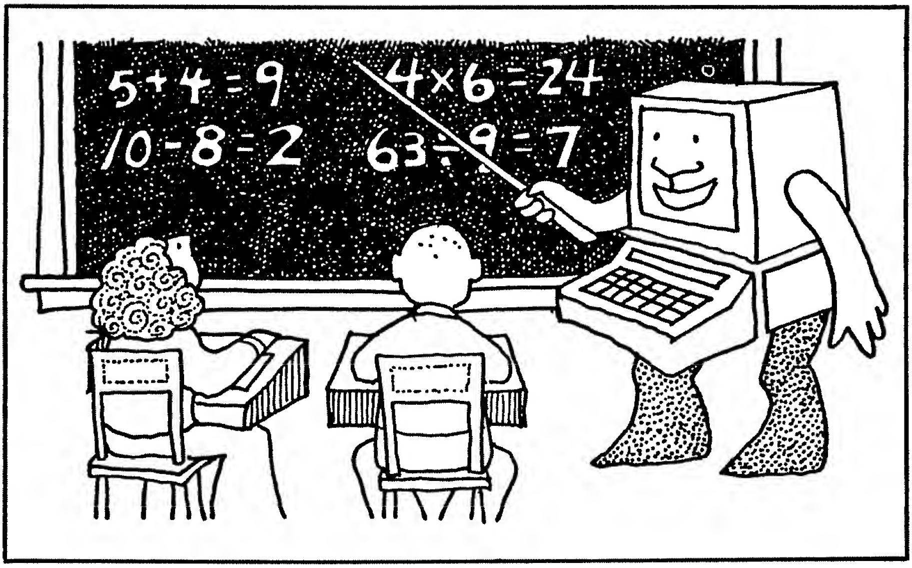
But there's a lot you can add. For example, when the text appears on the screen, the computer pauses (on line 4510) for a count of from 1 to 3000. Why not change that and make the computer display a certain screenful of text until you press the **ENTER** button. That way, you can relate how fast the computer jumps to a new screen to the speed of the reader. This would be especially helpful to younger children who still read very slowly.

And how about a subroutine that takes the report and stores it on a tape or disk? Your child might spend a long time typing in a report and might want to save sections of it at one time on tape or disk to work on later.

And how about printing the report? Do you have a printer? If you do, why make your child copy the whole report? Instead, add a subroutine that lets the child review the report on the TV screen and then type it out automatically on a printer.

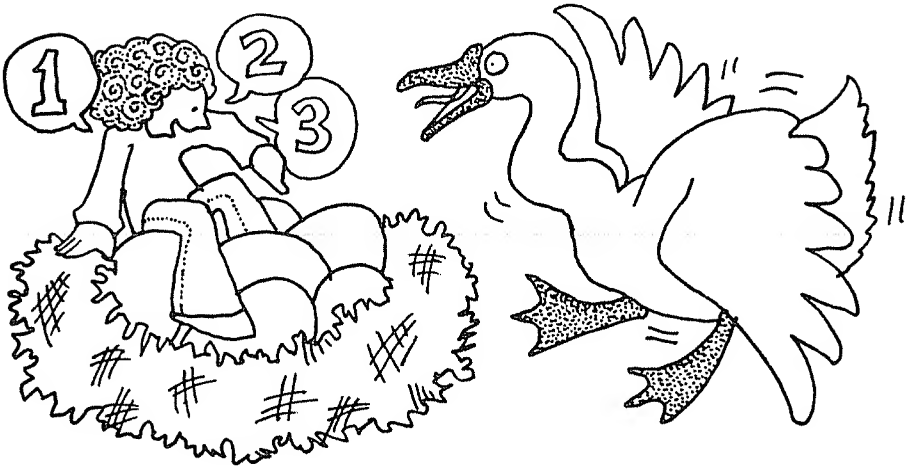
There are a lot of things you can dc with this program. With a few additions, it can turn your computer into the family's electronic typewriter.

NUMBERS



13

GOOSE EGGS



For Parents and Teachers ...

This game helps younger children learn how to count. It also teaches them where to find the number keys on the computer keyboard.

For Kids ...

Pretend you live on a farm. You go out to the goose coop. You find 1 goose egg in the goose nest.

The mama goose gets on the nest and lays a new egg.

That's 2 goose eggs.

Then the mama goose gets on the nest and lays another egg.

That's 3 goose eggs.

The mama goose keeps laying new eggs. You keep counting.

How many eggs can she lay?

How many eggs can you count?

The Game . . .

Program Name: **GOOSE**

```
50 REM *** COUNTING GOOSE EGGS
62 CLS
65 PRINT@33,"***  COUNT THE GOOSE EGGS
***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 CLS
100 FOR Y=2 TO 11:FOR X=6 TO 24 STEP 2
105 COUNT=COUNT+1
110 PRINT@ Y*32+X, "0";
120 PRINT@418,"HOW MANY GOOSE EGGS";:INP
UT E$
125 PRINT@418," "
130 IF VAL(E$) <> COUNT THEN 140
135 GOSUB 1010:REM * RIGHT ANSWER
137 PRINT@461," "
138 GOTO 150
140 GOSUB 2010:REM * WRONG ANSWER
142 PRINT@461," "
146 GOTO 120
150 NEXT X:NEXT Y
160 CLS:END
```

Highlights:

This is a short and simple program.

On line 100 you set up FOR-NEXT loops to position the eggs (number 0's) on the TV screen. X stands for the column. Y stands for the row. You place the eggs two spaces apart to make them easy to count. You put the X loop **inside** the Y loop to make the new eggs appear column by column (from left to right) across the screen.

On line 120 the computer asks how many eggs the goose has laid. After you have answered, on line 125 the computer erases the question and answer.

If you guess correctly, the computer jumps to the Happy Face subroutine on line 1010.

If you guess wrong, the computer jumps to the Sad Face subroutine on line 2010.

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Variables . . .

X Loop counter—column on TV screen for drawing the goose egg.

Y Loop counter—row on TV screen for drawing the goose egg.

COUNT Counter—how many goose eggs are on the screen.

E$ Your guess.

Do-It-Yourself . . .

This is a very simple program. You can add lots of bells and whistles. For example, you can make the goose eggs out of the special graphics. (See pages 250 to 266 in *Getting Started with Color BASIC.)*

Or you can have the computer make a noise each time the goose lays a new egg.

Or you can even draw a picture of the goose out of graphics.

14

SCRAMBLED EGGS

For Parents and Teachers ...

In the last chapter your children counted goose eggs along with the computer. But did you notice that the computer always counted the eggs **in order**?

In this chapter the computer scrambles the goose eggs. It has the option of drawing anywhere from zero goose eggs to 24 goose eggs on the TV screen. The children have to count the eggs and type in the correct answer. Then the computer erases the old eggs and draws a bunch of new ones.

The Game ...

Program Name: **GOOSE2**

```
50 REM *** COUNTING GOOSE EGGS
62 CLS
```

```
65 PRINT@33,"***  COUNT THE GOOSE EGGS
***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 CLS
100 COUNT=0
101 RANDOM=RND(25)-1
103 IF RANDOM=0 THEN 220
105 FOR Y=7 TO 8:FOR X=4 TO 26 STEP 2
107 IF COUNT=>RANDOM THEN 120
110 PRINT@ Y*32+X, "0";
115 GOSUB 3010
120 COUNT=COUNT+1
150 NEXT X:NEXT Y
220 PRINT@418,"HOW MANY GOOSE EGGS";:INP
UT E$
225 PRINT@418," "
230 IF VAL(E$)<>RANDOM THEN 240
232 PRINT@172,RANDOM;"EGGS"
235 GOSUB 1010:REM * RIGHT ANSWER
237 FOR PAUSE=1 TO 500:NEXT PAUSE
238 GOTO 90
240 GOSUB 2010:REM * WRONG ANSWER
242 PRINT@461," "
246 GOTO 220
3000 REM *** EGG SOUND
3010 SOUND89,1
3030 FOR K=1 TO 500:NEXT K
3050 RETURN
```

Highlights ...

Be sure to LOAD the Happy and Sad routine (FACES) before typing in this program.

This program is very similar to the GOOSE program. For details, take a look at the last chapter.

In the old program, GOOSE, we got the computer to count the eggs in order (1, 2, 3, and so on) by using a FOR-NEXT loop. In the new program, GOOSE2, we get the computer to choose the number of eggs at random by using the RND function on line 101.

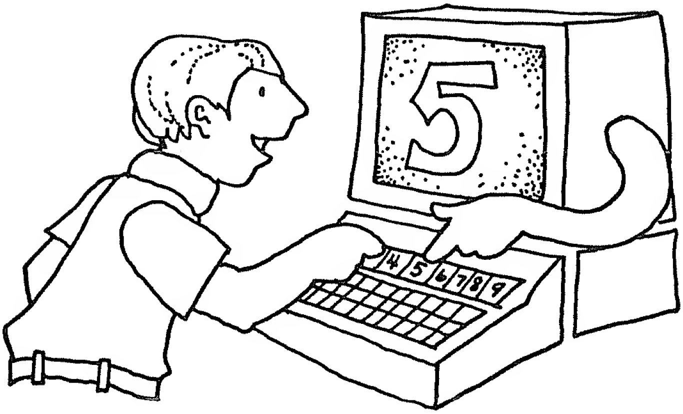
If the computer has chosen to print zero eggs (RANDOM equals 0), then, on line 103 we have the computer jump around the "print-eggs" loop and immediately ask how many eggs it has printed. (Since it has printed no eggs, the answer is 0).

The FOR-NEXT loops on lines 105 to 150 print the eggs in the center of the TV screen. Each time the computer "lays" an egg, it calls a sound subroutine on line 3010. The computer keeps count of how many eggs it has printed by using the variable COUNT. The computer stops printing eggs when COUNT reaches RANDOM (line 107).

In this new program, if you count the number of eggs correctly, the computer prints that number above the rows of eggs on the TV screen. Then (still displaying the eggs and the number) the computer prints the happy face. The happy face disappears and the computer erases the screen and draws a new bunch of eggs for you to count.

15

THE NUMBER RACE



For Parents and Teachers ...

This program helps children learn numerals and number sequence. It also helps them learn the location of the numerals on the computer keyboard.

For Kids ...

How fast can you find a number? Can you beat the computer?

In this game, the computer flashes a number on the picture screen. The number is between 0 and 9. As soon as the number appears, the computer begins counting. If you can punch the number button on the keyboard before the computer is done, you win. If not, the computer wins.

You win the first round if you can beat the computer for all ten numbers (0 to 9). But the computer's not beaten yet. It will play you four more rounds, each faster than the last. If you beat the computer all five rounds, it quits and you are the winner!

The Game ...

Program Name: **FINDNUM**

```
50 REM *** FIND THAT NUMBER!
60 CLS
62 PRINT@34,"***  FIND THAT NUMBER!  ***
"
64 FOR PAUSE=1 TO 1000:NEXT
70 MAX=300
120 FOR NUMBER=48 TO 57
125 COUNT=0:PRINT@271," "
126 PRINT@271,CHR$(NUMBER):SOUND 218,1
128 COUNT=COUNT+1:IF COUNT=MAX THEN 160
130 A$=INKEY$:IFA$=""THEN 128
140 A=ASC(A$)
150 IF A=NUMBER THEN GOSUB 1010:GOTO 170
160 GOSUB 2010:GOTO 125
170 NEXT NUMBER
171 IF MAX<=50 THEN 180
172 PRINT@386,"FASTER";:INPUT A$
173 PRINT@386," "
174 IF LEFT$(A$,1)<>"Y" THEN 177
175 MAX=MAX-50:GOTO 120
177 IF LEFT$(A$,1)<>"N" THEN 172
180 CLS:END
```

Highlights ...

Be sure to LOAD the Happy and Sad routine (FACES) before typing in this program.

This program is a cousin of Find the Capital Letter. For a detailed description of the program, look at Chapter five.

The major difference between this program and the other is that the FOR-NEXT loop counter, NUMBER, varies from 48 to 57. These are the ASCII codes for the numbers 0 to 9

The keyboard is opened with an INKEY$ command on line 130. Then, when a person presses one of the number buttons, the number is stored as a string in variable A$.

Line 150 tests for a match between the number the computer flashed on the screen and the person's answer. Let's say that this is the first time through the main loop, so NUMBER = 48. If the person typed a 0, a 48 (the ASCII code for 0) would be stored in variable A. On line 150, then, the match question would translate as:

```
IF 48=NUMBER
```

Back on line 120, NUMBER = 48, so the question translates to:

```
IF 48=48
```

Since this is true, that means the computer flashed a 0 on the screen, and the person pushed the 0 button before the computer finished counting. The person wins!

Variables ...

MAX — How high the computer counts each round. On the first round, the computer counts to 300. On the fifth round, the computer only counts to 100.

A$ — For YES (Y) or NO (N) answers to the computer's question "FASTER?"

NUMBER — FOR-NEXT loop counter—represents the ASCII code for the numbers 0 to 9.

COUNT — Counter—how high the computer has counted after flashing the number on the screen; varies between 1 and MAX.

A — Stores the internal keyboard code of the button the person pushes.

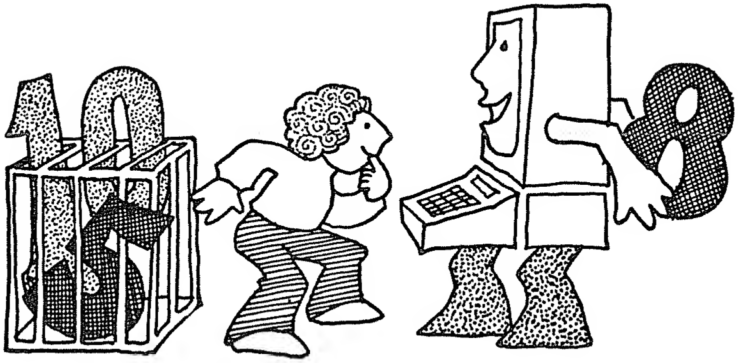
Do-It-Yourself . . .

The program uses the normal Happy and Sad (FACES) subroutine. You might modify the subroutines to make the computer print different messages if it won or lost. For example, if you won, the computer might say, "WOW! THAT WAS FAST!" or something similar. If you lost, it might say, "YOU SLOWPOKE! THAT WASN'T EVEN MY FASTEST!"

You can add PRINT commands to the front of the program to make the computer challenge the person to a button-pushing duel. You can add some PRINT commands at the end of the program to congratulate a person if he or she wins all four rounds and beats the computer.

16

GUESS MY NUMBER



For Parents and Teachers ...

This game helps children learn numbers and number sequence. It helps them become familiar with and adept at manipulating the "greater-than" and "less-than" relationship among numbers. It helps them learn efficient techniques for searching through a group of numbers.

For Kids ...

In this game, the computer "thinks up" a number between 1 and 20. It asks you your name. (Let's say your name is Hatty.) "HATTY," the computer says, "I AM THINKING OF A NUMBER BETWEEN 1 AND 20. WHAT IS IT?"

You try to guess the computer's number. Let's say you guess "10."

The computer might make a sad face and say: "SORRY! TOO HIGH! TRY AGAIN!"

You guess the number that is smaller than your last number. You guess "5."

The computer makes another sad face. "TOO HIGH!" it says.

You try again. You guess a number that's even smaller. You guess "3."

The computer makes a happy face. "RIGHT!" it says. It plays a happy tune.

The computer asks if you want to play again. You type "Y" or "YES" and the computer thinks up a new number for you to guess.

Here's a hint to help you guess numbers really fast: always pick a number about halfway between the highest possible number and the lowest possible number. This seems like a strange way to play the game, but it really works.

For example, in the beginning, the computer thinks up a number between 1 and 20. So you should think up a number halfway between 1 and 20. Your number should be 10.

Let's say the computer says 10 is too low. That means the computer's number must be between 11 and 20. This time you need to pick a number halfway between 11 and 20. Your number should be 16.

Let's say the computer says 16 is too high. That means the computer's number must be between 11 and 15. You need to pick a number halfway between 11 and 15. Your number should be 13.

The computer says: "RIGHT!" You guessed the number!

Now you try this trick. It's good for finding a single number buried in a lot of numbers—like finding a needle hidden in a haystack.

The Game ...

Program Name: **GUESS**

```
50 REM *** NUMBER GUESSING GAME
60 CLS
```

```
65 PRINT@35,"***  GUESS MY NUMBER!  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 CLS
102 PRINT" WHAT IS YOUR NAME";:INPUT N$
105 N=RND(20)
110 CLS
120 PRINT" ";N$;", I AM THINKING OF A NU
MBER"
130 PRINT" BETWEEN 1 AND 20."
140 PRINT
145 FOR PAUSE=1 TO 1000:NEXT PAUSE
150 PRINT @224," WHAT IS IT";:INPUT G$
160 G=VAL(G$):IF G=N THEN GOSUB 1010:GOT
O 200
170 GOSUB 2010:PRINT:GOTO 150
200 CLS
210 PRINT" WANT TO PLAY AGAIN, ";N$;:INP
UT A$
220 IF LEFT$(A$,1)="Y" THEN 105
230 IF LEFT$(A$,1)<>"N" THEN 200
240 CLS:END
```

Typing Hints ...

To get the reverse video words (which appear as lower case in the listing) on lines 2011, 2013, and 2085, you should press the SHIFT and 0 keys. To turn off the reverse video, press the SHIFT and 0 keys again.

Highlights ...

The key command in this game is the command on line 105 with the RND function. The RND function randomly selects a number between 1 and 20 and stores it in N.

The children's guess is INPUT into G$ on line 150. On line 160, the computer checks for a match between G and N. If the numbers match, the computer calls the Happy Face subroutine. If not, the computer calls the Sad Face subroutine.

For this game, instead of loading FACES, load in only the Happy Face subroutine before typing in the program.

The Sad Face subroutine has been greatly modified. Type in lines 2010 to 2110:

```
  2010 IF G<N THEN 2013
➧ 2011 PRINT@427,"too high!!"
  2012 GOTO 2085
➧ 2013 PRINT@427,"too low!!"
➧ 2085 PRINT@459,"try again!"
  2100 FOR PAUSE=1 TO 1000:NEXT PAUSE
  2110 CLS:RETURN
```

All SOUND commands and the sad face have been erased. Children are sure to guess the wrong number several times before guessing the right number. Hearing the sad sound and seeing the sad face so many times would be irritating and discouraging.

Variables . . .

PAUSE	Delay loop counter.
A$	Answer to the question "WANT TO PLAY AGAIN?"
N$	Your name.
N	Computer's mystery number.
G	The value of your guess.

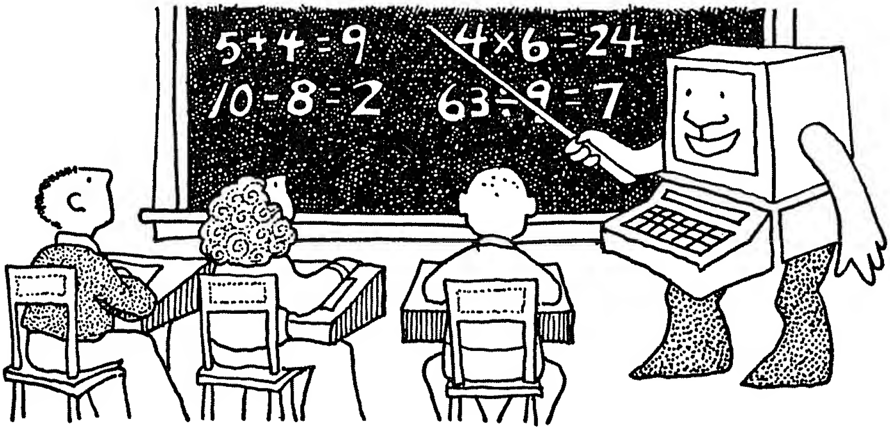
Do-It-Yourself . . .

For younger children you can shrink the range of numbers chosen by the computer. You need to change the PRINT command on line 130 and the RND function on line 105.

For older children you might want to expand the range of numbers the computer selects. You might also want to think up new categories of numbers for the children to guess; for example, even numbers, odd numbers, prime numbers, numbers only divisible by 11, squared numbers, square roots, etc. Numbers have personalities, just as people do. The Guess My Number game can teach children those personalities.

17

THE ARITHMETIC GAME



For Parents and Teachers ...

This game helps children learn addition, subtraction, multiplication, and division.

For Kids ...

When I was a child, I used to work on an arithmetic problem and then get frustrated because I didn't know if my answer was right or wrong. I wanted to know the answer right away. But I had to wait until the teacher graded my homework paper or my test. Sometimes that took days or even weeks. By the time I got my paper back, the problem was no longer fresh in my mind. If I got it right, I didn't know why. If I got it wrong, I didn't know why. And I no longer really cared.

Now you can solve arithmetic problems and learn instantly if you got them right or wrong. How? You can play the Arithmetic Game. It's like hiring the computer as your own private arithmetic teacher.

When you RUN The Arithmetic Game, the computer displays a "menu" on the TV screen. If you press a **1**, the computer challenges you with five addition problems. If you press a **2**, it throws five subtraction problems at you. If you press a **3**, you get five multiplication problems. If you press a **4**, you get five division problems.

The computer displays the problem on the screen. Then you try to think up the answer. If you want to, get out a piece of scrap paper and a pencil. That's what I do when I'm working on arithmetic problems.

Think up your answer to the problem, and then type it in. If you want to change your answer, press the left-arrow (←) key. When your answer is on the screen, press the **ENTER** button.

If you get the answer right, the computer draws a happy face, says "RIGHT!", and plays a happy tune.

If you get the answer wrong, the computer draws a sad face, groans, and encourages you to "TRY AGAIN!"

The Game ...

Program Name: MATH

```
50 REM *** THE ARITHMETIC GAME
60 CLS
65 PRINT@2,"*** THE ARITHMETIC GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 CLEAR 1000
72 B$=" "
100 GOSUB 3010:REM * MENU
170 FOR L=1 TO 5
180 N1=RND(10)
190 N2=RND(10)
200 GOSUB 3510:REM * GAME
250 NEXT L
265 PRINT@448,"  PLAY AGAIN <Y/N>";:INPU
T A$
```

```
270 IF A$="Y" THEN 100
280 IF A$ <> "N" THEN 265
300 CLS:END
3000 REM *** MENU SUBROUTINE
3010 CLS
3020 PRINT@102,"1. ADDING"
3030 PRINT@134,"2. SUBTRACTING"
3040 PRINT@166,"3. MULTIPLYING"
3050 PRINT@198,"4. DIVIDING"
3060 PRINT@256,"WHICH NUMBER (1, 2, 3, O
R 4)";:INPUT B$
3070 B=ASC(B$):IF B >= 49 AND B <= 52 TH
EN 3090
3080 GOTO 3010
3090 B=B-48:RETURN
3500 REM *** ARITHMETIC SUBROUTINES
3510 CLS
3513 N$=""
3515 ON B GOSUB 6010,6110,6210,6310
3520 INPUT N$
3600 ON B GOSUB 7010,7110,7210,7310
3605 IFW=VAL(N$) THEN GOSUB 1010:CLS:GO
TO 3650
3610 GOSUB2010:GOTO3510
3650 RETURN
6000 REM *** ADDING SUBROUTINE
6010 CLS:PRINT@138,"ADDING..."
6020 PRINT@199,N1;" + ";N2;" = ";
6030 RETURN
6100 REM *** SUBTRACTING SUBROUTINE
6110 CLS:PRINT@136,"SUBTRACTING..."
6120 IFN2>N1 THEN W=N2:N2=N1:N1=W
6130 PRINT@199,N1;" - ";N2;" = ";
6140 RETURN
6200 REM *** MULTIPLYING ROUTINE
6210 CLS:PRINT@136,"MULTIPLYING..."
6220 PRINT@199,N1;" X ";N2;" = ";
6230 RETURN
6300 REM *** DIVIDING SUBROUTINE
6310 PRINT@138,"DIVIDING..."
6320 W=N1*N2:N2=W
6330 PRINT@199,N2;" / ";N1;" = ";
6340 RETURN
7000 REM *** ADDING SUROUTINE
```

```
7010 W=N1+N2
7020 RETURN
7100 REM *** SUBTRACTING SUBROUTINE
7110 W=N1-N2
7120 RETURN
7200 REM *** MULTIPLYING SUBROUTINE
7210 W=N1*N2
7220 RETURN
7300 REM*** DIVIDING SUBROUTINE
7310 W=N2/N1:N2=W
7320 RETURN
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Highlights ...

The game begins (on line 100) by calling the "menu" subroutine beginning on line 3000. The subroutine lets you choose addition, subtraction, multiplication, or division.

The program uses a FOR-NEXT loop on lines 170 to 250 to control the number of the problems the computer "thinks up." Right now, the computer comes up with five problems per game. Then it asks if you want to "PLAY AGAIN?" If you type "Y" the computer calls the menu subroutine. You can continue solving the same kind of problems (addition, subtraction, multiplication, or division), or you can try something new.

The RND functions on lines 180 and 190 are set to randomly choose two numbers between 1 and 10. These are the numbers added together, subtracted, multiplied, or divided.

The actual arithmetic game is a subroutine beginning on line 3500. It is called five times by a GOSUB command on line 200.

The arithmetic subroutine calls two special purpose subroutines, depending on which kind of arithmetic you have chosen (addition, subtraction, multiplication, or division).

The first subroutine (beginning on lines 6000, 6100, 6200, or 6300) displays the arithmetic problems on the screen. The second subroutine (beginning on lines 7000, 7100, 7200, or 7300) calculates the correct answer to the problem so that it can be matched with your answer.

If you type a number and then change your mind, you can press the left-arrow (←) key to erase the number and start again.

Variables ...

PAUSE	Delay loop counter.
B$	Prints empty spaces (for erasing the screen).
A$	Your answer to the question "PLAY AGAIN?"
N$	Stores your answer to the arithmetic problem.
L	Loop counter—main loop.
N1	First random number for arithmetic problem.
N2	Second random number for arithmetic problem.
B	Which type of arithmetic problems you choose (addition, subtraction, multiplication, or division).
X	Location on the TV screen where your answer will appear.
W	The computer's answer to the arithmetic problem.

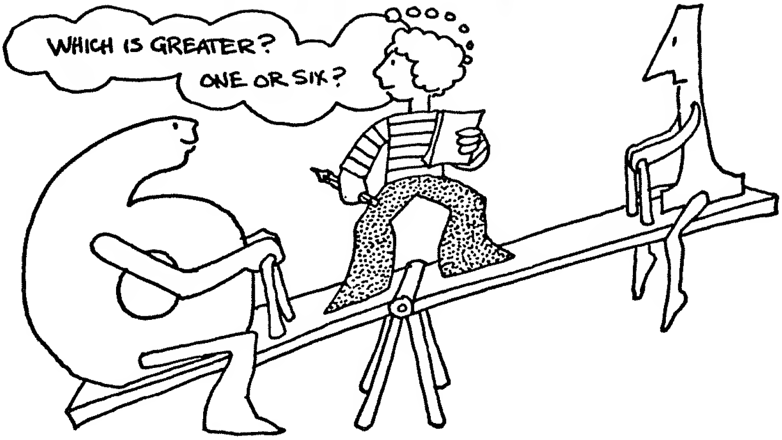
Do-It-Yourself ...

The program does not keep track of how many incorrect answers a child enters. It would be good to create an incorrect answer COUNT variable. Increment the variable (COUNT =

COUNT + 1) each time the child makes a wrong answer. When COUNT = 3, have the computer display the correct answer on the TV screen. Then have the program ask the child the same problem again.

18

GREATER THAN WHAT?



For Parents and Teachers ...

This game helps children learn the size relationships between numbers. It shows them two numbers, and then asks them if the first number is greater than the second, less than the second, greater than or equal to the second, or less than or equal to the second.

The game helps children learn the number comparison symbols: > (greater than), < (less than), >= (greater than or equal to), and <= (less than or equal to).

For Kids ...

Numbers describe things—apples, dragons, mudpies, and worms. The bigger the number, the more things. For example, if a blue room has 1000 worms in it and a red room has 2 worms in it, which room has more worms in it? The blue room, right?

That's because a 1000 is **greater than** 2. You can compare the numbers by drawing a thousand worms and then two more worms. Or you can take a shortcut and write:

$$1000 > 2$$

The > symbol means **greater than.** A thousand is greater than two.

You can also write the numbers like this:

$$2 < 1000$$

The < symbol means **less than.** Two is less than a thousand.

But what happens if you are out exploring an enchanted forest some afternoon after school, and you discover two caves? You turn on your pocket flashlight, enter one cave, and find 14 dragons. You back out carefully, enter the second cave and find 14 more dragons. You sneak out of the second cave and run all the way home.

You call the police and the fire department. Moments later, they are at your front door. They ask you what you saw.

To describe what you just saw, you take out a piece of paper and draw 14 dragons and then 14 more dragons.

You look at the cave pictures with all the dragons. Which cave had more dragons? Neither. They both had the same number of dragons. You can draw a big = (equal sign) between the two pictures of dragons. Or you can take a shortcut and write:

$$14 = 14$$

But what if you change your mind? In the first cave (the dirty cave), you think you saw shadows of other dragons. That means the dirty cave might have more dragons than the clean cave. You know that the clean cave had only 14 dragons. But some extra dragons might have been hiding in the dirty cave, behind the dragon garbage and the piles of old bones and dirty treasure you saw.

Now you can't say how many dragons were in the dirty cave. You use the symbol X to represent this mystery number. Is there anything you can say about X?

Yes. You know X is at least 14 and maybe more. You can write this as:

X >= 14

This means that X is **greater than or equal to** 14. You can also write this as:

14 <= X

This means 14 is **less than or equal to** X (the number of dragons in the dirty cave).

The police and fire fighters are satisfied with your description. They fly in a dragon S.W.A.T. team from China. The team comes equipped with lots of silk dragon nets. You lead the team back to the caves. You hope that there aren't too many more dragons in the dirty cave.

After you return from dragon hunting, sit down at the computer and play the Greater Than What? game. The computer flashes two numbers on the screen. You have to decide if one number is greater than the other; or less than the other; or greater than or equal to the other; or less than or equal to the other.

Think of the numbers as dragons.

In this game the computer expects a yes or no answer. You give a "YES" answer by pressing the **ENTER** button. You give a "NO" answer by pressing the **SPACE bar**.

The Game . . .

Program Name: **GREATER**

```
50 REM *** GREATER-THAN GAME
60 CLS
65 PRINT@1,"*** THE GREATER-THAN GAME **
*"
```

```
66 FORPAUSE=1TO1000:NEXT PAUSE
100 GOSUB 3010:REM * MENU
170 FORL=1TO5
180 N1=RND(10)
190 N2=RND(10)
200 ON B GOSUB 3510,4010,4510,5010
250 NEXT L
260 CLS
265 PRINT@416,"  PLAY AGAIN <Y/N>";:INPU
T A$
270 IF A$="Y" THEN 100
280 IF A$ <> "N"THEN 265
300 CLS:END
3000 REM *** MENU SUBROUTINE
3010 CLS
3020 PRINT@98,"1. GREATER THAN..........
.>"
3030 PRINT@130,"2. LESS THAN............
..<"
3040 PRINT@162,"3. GREATER THAN OR EQUAL
..>="
3050 PRINT@194,"4. LESS THAN OR EQUAL...
..<="
3060 PRINT@256,"WHICH NUMBER (1, 2, 3, O
R 4)";:INPUT B$:B=VAL(B$)
3070 IF B>=1 AND B<=4 THEN 3090
3080 GOTO 3060
3090 RETURN
3200 REM *** INITIALIZE SCREEN
3210 CLS
➽ 3220 PRINT@132,"enter=yes    space=no"
3230 RETURN
3500 REM *** GREATER-THAN (>)
3510 GOSUB 3210
3515 PRINT@71,N1;" > ";N2;" ?"
3520 K$=INKEY$:IF K$=""THEN 3520
3522 K=ASC(K$):IF K <> 13 AND K <> 32 TH
EN 3520
3530 IF (N1>N2 AND K=13) OR (N1<=N2 AND
K=32) THEN GOSUB 1010:GOTO 3570
3540 GOSUB 2010:GOTO3510
3570 RETURN
4000 REM *** LESS-THAN (<)
4010 GOSUB 3210
```

```
4015 PRINT@71,N1;" < ";N2;" ?"
4020 K$=INKEY$:IF K$=""THEN 4020
4022 K=ASC(K$):IF K <> 13 AND K <> 32 TH
EN 4020
4030 IF (N1<N2 AND K=13) OR (N1>=N2 AND
K=32) THEN GOSUB 1010:GOTO 4070
4044 GOSUB 2010:GOTO 4010
4070 RETURN
4500 REM *** GREATER THAN/EQUAL (>=)
4510 GOSUB 3210
4515 PRINT@71,N1;" >= ";N2;" ?"
4520 K$=INKEY$:IF K$="" THEN 4520
4522 K=ASC(K$):IF K <> 13 AND K <> 32 TH
EN 4520
4530 IF (N1>=N2 AND K=13) OR (N1<N2 AND
K=32) THEN GOSUB 1010:GOTO 4570
4550 GOSUB 2010:GOTO 4510
4570 RETURN
5000 REM *** LESS THAN/EQUAL (<=)
5010 GOSUB 3210
5015 PRINT@71,N1;" <= ";N2;" ?"
5020 K$=INKEY$:IF K$=""THEN 5020
5022 K=ASC(K$):IF K <> 13 AND K <> 32 TH
EN 5020
5030 IF (N1<=N2 AND K=13) OR (N1>N2 AND
K=32) THEN GOSUB 1010:GOTO 5070
5040 GOSUB 2010:GOTO 5010
5070 RETURN
```

Typing Hints ...

The words on line 3220 are in lower case. This means they are to be typed in reverse video. To get the words to appear in reverse video, press the **SHIFT** and **0** keys simultaneously. To turn off the reverse video, press the **SHIFT** and **0** keys again.

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Highlights . . .

This game is similar to The Arithmetic Game. When the program begins, it calls the menu subroutine beginning on line 3000. You can choose four different kinds of number comparison problems: greater than (>), less than (<), greater than or equal (>=), or less than or equal (<=).

The FOR-NEXT loop on lines 170 to 250 is the main program loop. It sets the number of problems per game. Right now the loop is set to offer you five problems per game.

The two RND functions on lines 180 and 190 randomly select the numbers to be compared. The ON B GOSUB command on line 200 calls the appropriate subroutine, based on your choice of problem type (greater than, less than, greater than or equal, less than or equal).

The four subroutines beginning at lines 3500, 4000, 4500, and 5000 handle the four types of problems. They display the problem on the TV screen (see the subroutine beginning on line 3200). They accept your answer and check to see if it is correct. If it is correct, they call the Happy Face subroutine. If it is incorrect, they call the Sad Face subroutine.

You give a "YES" answer by pressing the **ENTER** button (an ASCII value of 13). You give a "NO" answer by pressing the SPACE bar (an ASCII value of 32).

Variables . . .

PAUSE	Delay loop counter.
A$	Answer to the question "TRY AGAIN?"
Y	Loop counter—main problem loop.
N1	First number the computer randomly selects.
N2	Second number the computer randomly selects.
B	Your choice of problem type (>, <, >=, or <=).
K	Your answer (ENTER = YES, SPACE bar = NO).

Do-It-Yourself . . .

You can add a new problem type—not equal (<>).

You can combine the Greater Than What? game and The Arithmetic Game. First the children would have to do an arithmetic problem. Then they would have to use one of the five number comparison symbols (>, <, >=, <=, or <>).

Or you can combine the two games by asking questions like:

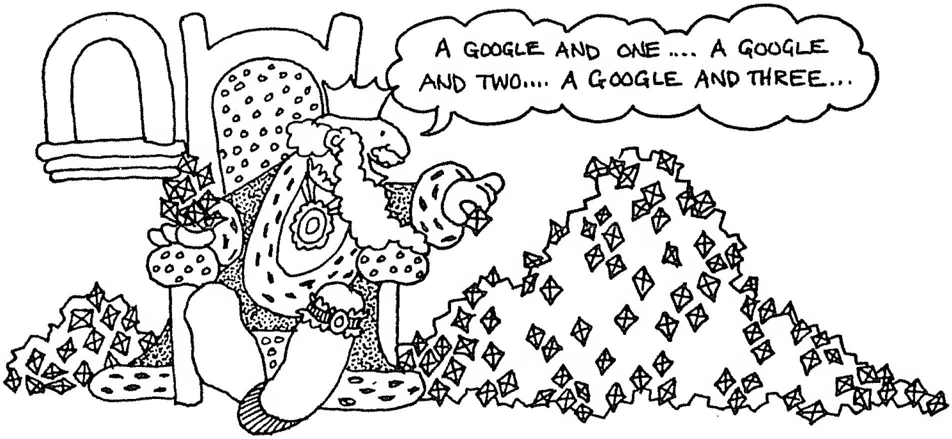
```
5 + 9 > 14 + 0 ?
```

Or you could help older children practice algebra with problems like:

```
IF X = 100 THEN IS X/5 >= .1X + 9 ?
```

19

THE RICH KING


A GOOGLE AND ONE .... A GOOGLE AND TWO.... A GOOGLE AND THREE...

For Parents and Teachers ...

This game helps children practice addition. It helps them sharpen their concentration and memory.

A rich king goes from room to room in his castle, counting his diamonds. The children follow the king through the different-colored rooms. They count the diamonds in each of the rooms. At the end they add up the diamonds in all the rooms.

For Kids ...

Once upon a time, there was a country named Cybernia. The king of Cybernia was a fat, little man. But he was very, very rich. All of his riches were in the form of diamonds. The king lived in a huge castle. And all the rooms of the castle were filled with diamonds. Each room was a different color, so the king could remember where he had been and not go back to the same room twice.

Every morning when the king woke up, the first thing he did was walk through all the rooms of his castle to count his diamonds. Every night just before the king went to bed, he put on his pajamas and slippers and brushed his teeth. Then he walked through all his castle rooms and counted his diamonds again.

At night the little king saw diamonds floating in his dreams.

The king had only one problem. He wasn't very good at counting. This made the king very nervous. Sometimes he was so nervous that he couldn't sleep at night, and so, he could not dream about his beautiful diamonds.

To make matters worse, every day the king got a new wagonload of diamonds from the diamond miners in his kingdom. And every day the king had to spend lots of his diamonds to pay his bills.

The king was getting very jittery and very sleepy. Then he came up with a solution. He would hire a diamond counter to help him count his diamonds. Whom did he hire?

You!

You are the king's diamond counter. It is your job to follow the little king through the rooms of his castle, counting the diamonds.

You can do this two ways. Either you can count the diamonds one at a time: 1, 2, 3, 4, and so on, through all the rooms. Or, you can count the diamonds in each room, write down the number on a piece of paper, and then add up the numbers for all the rooms to get the total number of diamonds.

Good luck! If you get the number wrong, the king looks sad and moans. He grabs your hand and leads you back through the rooms in the castle to count the diamonds again.

Each time you get the number of diamonds right, the king smiles at you, shouts "RIGHT!" and plays a happy tune on his royal whistle. Maybe someday he will even give you a bag of diamonds as a reward!

The Game ...

Program Name: **KING**

```
5 REM *** COUNTING DIAMONDS
10 REM PROGRAM AUTHORS:
20 REM BETH ANN HOSTUTLER AND
30 REM JONI BURDETTE
40 CLEAR 500
60 CLS
65 PRINT@37,"*** THE RICH KING ***"
69 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM P(15):DIM K(8):DIM ROOM(8)
80 FOR L=1 TO 15:READ P(L):NEXT L
85 DIM C$(8),C(8)
90 FOR I=1 TO 8
92 READ C$(I):READ C(I):NEXT I
94 FOR I=1 TO 8:K(I)=0:NEXT I
100 REM GENERATE NUMBER OF
105 REM DIAMONDS (1-50)
110 DIAMONDS=RND(50)
112 DT=0
115 FOR I=1 TO 8
120 R=RND(15)
122 J=RND(8): IF K(J)<>0 THEN 122
124 K(J)=I
125 IF DT+R> DIAMONDS THEN R=DIAMONDS-DT
135 ROOM(I)=R:DT=DT+R
140 NEXT I
155 DT=0: REM DIAMOND TOTAL
200 J=1
202 CLS K(J)
227 REM *** DRAW KING
240 FOR L1=5 TO 15:FOR L2=1 TO 12
242 PRINT@L1*32+L2,CHR$(128);:NEXT L2:NE
XT L1
250 PRINT@166,CHR$(178);CHR$(178);CHR$(1
78);
252 PRINT@197,CHR$(186);:PRINT@200,CHR$(
182);
253 PRINT@198,CHR$(178);:PRINT@230,CHR$(
184);
254 PRINT@229,CHR$(186);:PRINT@232,CHR$(
180);CHR$(184);
256 PRINT@260,CHR$(178);CHR$(128);CHR$(1
88);CHR$(188);CHR$(184);
258 PRINTCHR$(178);
260 PRINT@289,CHR$(165);CHR$(175);CHR$(1
75);
262 PRINT@293,CHR$(200);CHR$(194);CHR$(1
94);CHR$(200);
264 PRINTCHR$(165);CHR$(175);CHR$(175);
266 PRINT@321,CHR$(165);CHR$(170);CHR$(1
75);
268 PRINT@326,CHR$(196);
270 PRINT@329,CHR$(165);CHR$(170);CHR$(1
75);
272 PRINT@353,CHR$(165);CHR$(170);CHR$(1
75);
274 PRINT CHR$(233);CHR$(233);CHR$(233);
CHR$(233);CHR$(233);
276 PRINT CHR$(173);CHR$(170);CHR$(175);
278 PRINT@385,CHR$(180);
280 PRINT@387,CHR$(175);
282 PRINT@393,CHR$(165);CHR$(170);CHR$(1
80);
284 PRINT@419,CHR$(175);
286 PRINT@425,CHR$(165);CHR$(170);
288 PRINT@450,CHR$(165);CHR$(175);
290 PRINT@457,CHR$(165);CHR$(175);
292 PRINT@482,CHR$(175);
294 PRINT CHR$(233);CHR$(233);CHR$(233);
CHR$(233);
296 PRINT CHR$(233);CHR$(233);CHR$(233);
298 PRINT CHR$(173);CHR$(170);
420 GOSUB 3000:REM DRAW DIAMONDS
435 IF DT<DIAMONDS THEN 437
436 GOTO 440
437 J=J+1:IF J<9 THEN 202
440 CLS
450 PRINT" HOW MANY DIAMONDS IN ALL ROOM
S":INPUT A$
455 REM IS PLAYER'S SUM CORRECT?
460 IF VAL(A$)=DT THEN 480
470 GOSUB 2010:REM SAD FACE
475 DT=0:GOTO 200
```

```
480 GOSUB 1010:REM HAPPY FACE
490 PRINT@481," ";
500 PRINT"PLAY AGAIN";:INPUT M$
510 IF LEFT$(M$,1)="Y" THEN 94
520 IF LEFT$(M$,1)<>"N" THEN 490
530 CLS:END
3000 REM *** DRAW DIAMONDS
3010 IF ROOM(J)=0 THEN 3157
3020 FOR L1=4 TO 9:FOR L2=15 TO 30
3022 PRINT@L1*32+L2,CHR$(128);:NEXT L2:N
EXT L1
3030 FOR I=1 TO ROOM(J)
3040 PRINT@P(I),CHR$(201);
3050 PRINT@P(I)+1,CHR$(198);
3060 PRINT@P(I)+32,CHR$(200);
3070 PRINT@P(I)+33,CHR$(196);
3080 NEXT I
3110 DT=DT+ROOM(J)
3157 PRINT@369,"COUNT DIAMONDS";
3160 PRINT@401,"IN ";C$(K(J));" ROOM";
3162 PRINT@465,"PRESS 'ENTER'";
3163 PRINT@497,"FOR NEXT ROOM";
3164 A$=INKEY$:IF A$="" THEN 3164
3165 RETURN
4000 DATA 214,217,211,208,220
4010 DATA 150,147,153,278,281
4020 DATA 275,144,156,272,284
5000 DATA GREEN,1,YELLOW,2,BLUE,3,RED,4,
BUFF,5
5010 DATA CYAN,6,MAGENTA,7,ORANGE,8
```

Background . . .

The Rich King game was designed by Joni Burdette and Beth Ann Hostutler, two students at Patrick Henry High School in Roanoke, Virginia.

The girls took the high-school computer course to escape from their other "boring" classes. They became so excited with computers that they began planning for a computer career. Both girls became enthusiastic computer gamers at a local game arcade.

Highlights ...

Be sure to LOAD in the Happy and Sad routine (FACES) before typing in the program.

This program reads eight colors into the array, C. These are the colors of the king's rooms. The color values and the color names are in the DATA commands on lines 5000 and 5010.

The major program loop is on lines 200 to 437. The room the king is in depends on the loop counter, J.

When we enter each room the king is by the door. This is accomplished by drawing a black background.

The king is drawn on lines 227 to 298.

The diamonds are drawn by the subroutine beginning on line 3000.

Variables ...

PAUSE Delay loop counter.

A$ Accepts **ENTER** button—to advance to king's next diamond room. Also holds your count (the number of diamonds in all of the king's rooms).

C$ Array—name of the color of the current diamond room.

C Array—stores the 8 color values (for up to 8 diamond rooms).

I Loop counter.

M$ Your answer to the question "PLAY AGAIN?"

DIAMONDS
The total number of diamonds in all of the king's rooms.

DT Total number of diamonds counted at any point in the game.

J Loop counter—main program loop (allows the king to visit up to 8 diamond rooms).

P Array—holds values used to position the diamonds.

K Array—holds the color value (1 to 8) of the current room.

R The number of diamonds in any one room.

ROOM Array—number of diamonds in the current room.

Do-It-Yourself . . .

The program puts a lot of diamonds in each room (anywhere from 1 to 15 diamonds). This results in the king visiting fewer rooms. You can change the RND function on line 120 to make the computer put fewer diamonds in each room. This way the king can visit more rooms, and the children won't have to keep track of so many diamonds in each room.

For younger children you can reduce the total number of diamonds by modifying the RND function on line 110.

Also, for younger children, you can have the king introduce himself at the beginning of the program—"HI! I'M THE RICH KING." You can have the king explain the game and advise the child to have a piece of paper to record the number of diamonds in each of the different-colored rooms. "When I am done walking through my castle," the king might say, "you can add up all the diamonds and tell me how rich I really am."

Finally, you might consider having the king draw the diamonds more slowly, so that very young children can add them up as they appear on the screen.

20

THE DISAPPEARING GHOSTS

For Parents and Teachers ...

This game helps children practice subtraction.

A number of ghosts appear above their tombstones. They disappear. Then some of the ghosts reappear. The game asks children how many of the ghosts disappeared and didn't return.

For Kids ...

Imagine that it is late at night. You and your best friend are late for dinner. You take a shortcut across an old cemetery.

You hear a noise. You both turn around.

You are surrounded by ghosts! They are everywhere, peeking over the tops of tombstones.

You start to scream!

The ghosts disappear.

Then some of them return!

This time you don't take the time to scream. You plan to do some quick disappearing yourself. You grab your friend's hand and run like the wind!

You return home, puffing and panting. You tore your pants scrambling over the cemetery fence. You look over your shoulder. No ghosts.

You sit down at your computer. You load in the Ghosts game and type RUN.

There they are again! Ghosts' heads pop over the top of 12 tombstones. You count them.

The ghosts disappear.

Then seven of the ghosts return.

The computer asks you: "HOW MANY GHOSTS THE FIRST TIME?"

You answer: "12."

The computer asks you: "HOW MANY GHOSTS THE SECOND TIME?"

You answer: "7."

Then the computer asks you: "HOW MANY OF THE GHOSTS DISAPPEARED?"

This time you kept a careful count. You type the number **5.**

A happy blue and white ghost appears on the TV screen. "RIGHT!" it says.

The Game ...

Program Name: **GHOSTS**

```
5 REM *** DISAPPEARING GHOSTS
10 REM ANGELA BRADSHAW
50 CLS
51 PRINT@34,"*** DISAPPPEARING GHOSTS **
*"
53 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 N=RND(25)
105 N2=RND(25)
106 IF N<N2 THEN Z=N:N=N2:N2=Z
```

```
107 O1=N:O2=N2
108 CLS 5
122 PRINT@226,"count the ghosts!";
125 COUNT=0
130 FOR PAUSE=1 TO 1000:NEXT PAUSE
165 N1=N
210 GOSUB 3005:REM PRINT GHOSTS
260 REM BOTH SETS PRINTED?
270 COUNT=COUNT+1
280 FOR DELAY=1 TO 3000:NEXT DELAY
305 IF COUNT=2 THEN 335
310 N=N2
318 CLS 5
319 PRINT@226,"here they are again";
320 FOR PAUSE=1 TO 1000:NEXT PAUSE:GOTO
210
335 CLS
337 PRINT@34,"how many ghosts the first
time"
340 INPUT E$
341 F$="A"
342 IF VAL(E$)=N1 THEN 344
343 GOTO 360
344 PRINT" how many ghosts the second ti
me"
345 INPUT F$:IF VAL(F$)=N2 THEN 347
346 GOTO 360
347 PRINT:PRINT" HOW MANY GHOSTS DISAPPE
ARED":INPUT G$
350 IF VAL(G$)=N1-N2 THEN 380
360 GOSUB 2010
370 N=O1:N2=O2:F=VAL(F$):IF F=0 OR F=N2
THEN 108
375 COUNT=1:GOTO 310
380 GOSUB 1010
400 CLS:PRINT@450,"PLAY AGAIN";:INPUT A$
410 IF LEFT$(A$,1)="Y" THEN 100
420 IF LEFT$(A$,1)<>"N" THEN 400
430 CLS:END
3000 REM PRINT GHOSTS
3005 CLS
3010 FOR X=3 TO 13 STEP 5
3012 FOR Y=1 TO 28 STEP 3
3014 P=X*32+1+Y
3020 PRINT@P,CHR$(142);
```

```
3022 PRINT CHR$(141);
3024 PRINT@P+32,CHR$(130);
3026 PRINT CHR$(129);
3030 NEXT Y
3040 NEXT X
3100 Y=1:X=2
3105 FOR L=1 TO N
3110 PRINT@Y*32+X,CHR$(206);
3120 PRINT CHR$(206);
3130 PRINT@Y*32+32+X,CHR$(206);
3140 PRINT CHR$(205);
3150 X=X+3:IF X>29 THEN X=2:Y=Y+5
3160 NEXT L
3200 RETURN
```

Typing Hints . . .

The words on lines 122, 319, 337, and 344 appear in lower case. This means they are to be typed in reverse video. To type the words in reverse video, press the **SHIFT** and **0** keys. To turn off the reverse video, press the **SHIFT** and **0** keys again.

Be sure to LOAD in the Happy and Sad routine (FACES) before typing in this program.

Background . . .

The Disappearing Ghosts game was designed by Angela Bradshaw, a student at Patrick Henry High School in Roanoke, Virginia.

In order to finish her program on schedule, Angela took the computer home from school with her. She spent an entire weekend teaching the computer how to draw the ghosts' shape just right.

When she returned to school on Monday morning she showed off her ghosts. Her teacher decided that the ghosts' shape still wasn't realistic enough. Angela had to rewrite the entire program.

Angela was disappointed. She almost gave up. But instead, she spent all day Monday creating a new Ghosts program.

On Tuesday Angela showed her teacher the new ghosts. He liked them! And Angela was proud of her new program. This is the program she created.

Highlights ...

The number of ghosts in the first group is chosen by the RND function on line 100. The number of the ghosts in the second group is chosen by the RND function on line 105.

The ghosts are drawn by the subroutine beginning on line 3000.

Variables ...

PAUSE	Delay loop variable.
DELAY	Delay loop variable.
A$	Your answer to the question "PLAY AGAIN?"
N	Number of the ghosts in the first group.
N2	Number of the ghosts in the second group.
Z	Temporarily stores the number of ghosts in the first group (only used when the number of ghosts in the first group is less than the number of ghosts in the second group—subtracting the second number from the first would result in a negative number, so the numbers are reversed).
O1	Value of N is stored in O1, in case the child gets the answer wrong and the ghosts have to be redrawn.
O2	Stores value of N2 in case ghosts need to be redrawn.

COUNT Counts number of groups of ghosts (total of two groups).

E$ Your answer—how many ghosts the first time?

F$ Your answer—how many ghosts the second time?

G$ Your answer—how many ghosts disappeared?

X Column position of current ghost being drawn on the TV screen.

Y Row position of the ghosts.

Do-It-Yourself ...

The program doesn't keep a count of the number of incorrect answers. You can create a variable, WRONG, and increment it (WRONG=WRONG+1) each time the child types in the wrong answer. Then if WRONG > 2, you can print the correct answer and have the computer make the same number of ghosts disappear again.

You might also consider adding sound effects to this program. You can start the program with eerie noises—wind whistling, howling wolves, and so forth. Then, each time a ghost appears, you can have the computer make a whooshing or popping noise.

You can also make the ghosts appear in a different way. The ghosts disappear now by having the computer erase the entire screen. Instead, you can redraw the ghosts using color 1. This will erase the ghosts one at a time. It will make it easier for the children to subtract the ghosts and get the right answer. It might also make the scene in the cemetery a little more realistic and a little scarier!

21

THE HAMBURGER CONTEST

For Parents and Teachers ...

This game helps children practice multiplication.

The computer draws from one to nine hamburgers and from one to nine people. It asks, for example, if four people eat three hamburgers each, how many hamburgers are eaten?

For Kids ...

Do you like hamburgers?

I hope so, because you've been elected by your school to go to the first national KIDS' HAMBURGER EATING CONTEST.

You go to Washington, D.C. You and eight other children sit down at a huge picnic table. On top of the table are nine stacks of juicy hamburgers. The stacks of burgers are so high that they reach into the sky.

The President of the United States comes to the table, wishes you all good luck, and fires a pistol into the sky. You and the other children start munching burgers.

Half an hour later, the President fires the pistol again. "Boy!" he says. "A lot of burgers were eaten here today. You each ate seven burgers, and there are nine of you. If nine people each eat seven burgers, how many burgers are eaten?"

You still have a half-eaten burger in your mouth. But you have the answer. You raise your hand. "MFFF!" you say.

"What was that?" the President asks.

You spit the hamburger out of your mouth. Now you can talk. "63!" you yell. "63 hamburgers."

"RIGHT!" says the President.

The Game . . .

Program Name: **BURGER**

```
5 REM *** HAMBURGER CONTEST
10 REM MACK MCGHEE
12 CLEAR500:DIM G(12)
50 FOR L=1 TO 9:READ HG(L):NEXT
52 FOR L=1 TO 9:READ HP(L):NEXT
55 FOR L=1 TO 9:READ P(L):NEXT
60 FOR L=1 TO 12:READ G(L):NEXT
100 CLS
105 PRINT@34,"*** HAMBURGER CONTEST ***"
110 FOR DLAY=1 TO 3000:NEXT
120 MAN=RND(8)+1
121 HAM=RND(8)+1
130 CLS 0:T=HAM:RC=0
132 FOR I=1 TO T
134 P=HP(I)
136 FOR IH=1 TO 9
138 PRINT@ P+RC, CHR$(HG(IH));
140 RC=RC+1:IF RC=3 THEN RC=0:P=P+32
142 NEXT:NEXT
200 REM DRAW PEOPLE
210 B=MAN:RC=0
220 FOR I=1 TO B
230 P=P(I)
```

```
240 FOR IH=1 TO 12
250 PRINT@ P+RC, CHR$(G(IH));
260 RC=RC+1:IF RC=4 THEN RC=0:P=P+32
270 NEXT:NEXT
300 PRINT@109,CHR$(169);:PRINT@113,CHR$(
166);:PRINT@142,CHR$(169);
310 PRINT@144,CHR$(166);:PRINT@175,CHR$(
175);:PRINT@206,CHR$(166);
320 PRINT@208,CHR$(169);:PRINT@237,CHR$(
166);:PRINT@241,CHR$(169);
500 REM *** QUIZ
510 GOSUB 800:REM CLEAR WINDOW
520 IF B=1 THEN M1$="PERSON EATS"
530 IF T=1 THEN M2$="BURGER, HOW"
540 IF B>1 THEN M1$="PEOPLE EAT"
550 IF T>1 THEN M2$="BURGERS EACH, HOW"
560 IF B>1 AND T=1 THEN M2$="BURGER EACH
, HOW"
562 IF B=1 AND T>1 THEN M2$="BURGERS, HO
W"
570 PRINT@302,"IF";B;M1$;T;
580 PRINT@334,M2$;
590 PRINT@366,"MANY BURGERS ARE";
600 PRINT@398,"EATEN?";
605 F$=""
610 A$=INKEY$:IFA$=""THEN610
612 IF ASC(A$)=13 THEN 650
614 IF ASC(A$)=8 THEN PRINT@404,"  ";:PR
INT@403,"?";:GOTO 605
620 IF ASC(A$)<48 OR ASC(A$)>57 THEN 610
622 IF LEN(F$)>1 THEN 610
630 F$=F$+A$:PRINTA$;:GOTO 610
650 FF=VAL(F$)
660 IF FF=B*T THEN CLS1:GOSUB 1010:GOTO
700
670 CLS1:GOSUB 2010
690 GOTO 130
700 PRINT@448,"PLAY AGAIN";:INPUT A$
710 IF LEFT$(A$,1)="Y" THEN 120
715 IF LEFT$(A$,1)<>"N" THEN 700
720 CLS:END
800 REM *** CLEAR WINDOW
810 B$="                  ":REM 18 SPACE
S
```

```
820 PRINT@302,B$;:PRINT@334,B$;:PRINT@36
6,B$;
840 PRINT@398,B$;
860 RETURN
900 DATA 147,147,147,181,191,186,156,156
,156
920 DATA 116,120,124,212,216,220,20,24,2
8
930 DATA 128,132,136,0,4,8,256,260,264
940 DATA 241,253,254,242,243,251,247,243
,241,254,253,242,241,250,245,241
```

Typing Hints . . .

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Background . . .

The Hamburger Contest game was designed by Mack McGhee, a student at Patrick Henry High School in Roanoke, Virginia.

Mack writes: "Working on this program was a small adventure for me. When I began taking computer science at the beginning of this year, I never dreamed that after only a couple months I would be writing a computer program for a book.

"After changing the program many times and completely starting over once, I noticed the deadline approaching. My teacher, David James, began taking us home from school at 5 or 6 every night. He carried the class's one computer to all the different kids' houses. Sometimes he would arrive with the computer at my house as late as 10:30 PM. Once, on a school night, I worked on my program until 2:30 in the morning.

"Doing this program was a good experience. I hope you have as much fun playing it as I did making it."

Highlights ...

The computer randomly selects the number of hamburgers and people with RND functions on lines 120 and 121. The number of people and hamburgers varies between one and nine.

On lines 132 to 142, the computer uses two loops (representing screen position and graphics) to draw the pictures of the hamburgers on the TV screen.

On lines 200 to 270, the computer uses two more loops (representing screen position and graphics) to draw the pictures of the people in the hamburger-eating contest.

Only a single hamburger and only a single person is actually plotted. But the loops vary the screen positions so that between one and nine copies of the hamburger and person are made on the screen.

Variables ...

DLAY	Delay loop counter.
A$	Your answer to the questions.
HAM	Random number of hamburgers selected by the computer.
MAN	Random number of people selected by the computer.
T	Stores same value as HAM.
B	Stores same value as MAN.
HP	Array—hamburger screen positions.
HG	Array—hamburger graphics.
G	Array—graphics for people.
P	Array—people screen positions.
IH	Loop counter—positions graphics for both burgers and people.

FF Your answer—the number of burgers that you think are eaten.

I Loop counter—the number of people and hamburgers drawn in the graphics loops.

Do-It-Yourself . . .

You can add a wrong-answer counter to the game. You can increment the counter by one each time a child gives the wrong answer. If a child gives the wrong answer three times in a row, you can have the computer print out the correct answer and then give the child the same problem again.

You can experiment with ways to make the color of the people or the hamburgers different.

You can add sound-effects to the game, such as the munching noises you might hear in a real hamburger contest.

How about making the people different shapes? You can have the computer draw fat faces and skinny faces, faces with long hair and short hair, and big noses and little noses.

22

PEPPERONI, PLEASE!

For Parents and Teachers ...

This game will help children practice their division and fractions.

The computer draws a pizza pie on the TV screen. It divides the pie into anywhere from two to eight slices. Then it draws a random number of people on the screen. For example, the computer might ask: If there are three people trying to eat six slices of pizza, how many slices does each person get?

When the child has correctly answered the question (or answers the question incorrectly three times in a row), the computer prints the correct answer (2 slices) and displays, as a fraction, the amount of the whole pizza each person gets (2/6 or 1/3).

For Kids . . .

Imagine that one night you and a bunch of your friends go to a pizza restaurant. You order the biggest pizza, but there are a lot of people, and everyone is starved.

The pizza arrives, and people attack it. But no one has figured out how much pizza each person should get. Everyone tries to grab as many slices of pizza as they can reach. Fights break out. People have pieces sticking out of their ears, their pockets, and their socks. The pizza is gone, and nobody got more than a couple of bites.

You go home and take a long shower. You use lots of soap and hot water to try to scrub the mozzarella cheese and mushrooms out of your hair.

Now imagine that you plan to go with these same friends to a pizza restaurant again tonight. This time you plan to be prepared. You sit down at your computer and RUN your new game, Pepperoni, Please!

The game draws a picture of a pizza on the TV screen. The pizza has between two and eight slices. Then the game draws anywhere from one to eight people on the screen. The game asks: "HOW MANY SLICES PER PERSON?" and "HOW MANY SLICES IN THE WHOLE PIZZA?"

You spend an hour playing the game. Now you're ready to go to the restaurant with your friends. No matter how many people go, and no matter how many slices of pizza there are, you will know how much pizza to give each person.

The Game . . .

Program Name: **PIZZA**

```
5 REM *** PEPERONI, PLEASE!
10 REM HOWARD BOGGESS
20 CLEAR 500
30 DIMG(12)
55 FOR L1=1TO8:READ P(L1):NEXT
60 FOR L1=1TO12:READ G(L1):NEXT
```

```
  100 CLS
  105 PRINT@39,"PEPPERONI, PLEASE!"
  115 PRINT@105,"AN EXERCISE IN"
  125 PRINT@171,"FRACTIONS"
  130 FOR DELAY=1 TO 6000:NEXT
  200 M=RND(7)+1
  201 N=M
  220 IF M=3 THEN N=M*RND(2)
  224 IF M=4 THEN N=M*RND(2)
  230 IF M = 2 THEN N=M*RND(4)
  240 GOSUB 7005:REM DRAW PEOPLE
  245 GOSUB 9010:REM DRAW PIZZA
  250 ON N GOSUB 0,3000,3500,4000,4500,500
  0,5500,6000
  260 REM * ABOVE DISPLAYS SLICES
» 300 PRINT@385,"count the people";
» 310 PRINT@449,"count the slices";
  320 FOR DELAY=1 TO 6000:NEXT
  322 PRINT@385,"                ";
  324 PRINT@449,"                ";
  326 SOUND 176,2
» 335 PRINT@385,"HOW MANY slices";
» 336 PRINT@449,"PER person?";
  337 A$=INKEY$:IF A$=""THEN337
  338 YY=VAL(A$):IF YY<1 OR YY>8 THEN SOUN
  D32,2:GOTO 337
  339 PRINT@461,YY;
» 340 PRINT@417,"IN THE whole    ";
  342 PRINT@449,"                ";
» 344 PRINT@449,"pizza?";
  345 SOUND176,2
  346 A$=INKEY$:IFA$=""THEN 346
  348 YZ=VAL(A$):IF YZ<1 OR YZ>8 THEN SOUN
  D32,2:GOTO 346
  350 PRINT@456,YZ;
  380 IF YY=N/M AND YZ=N THEN 395
  382 CLS:GOSUB 2010
  383 ERROR = ERROR +1
  384 IF ERROR>2 THEN ERROR=0:GOTO 397
  390 GOTO 240
  395 CLS:GOSUB 1010:REM HAPPY FACE
  397 GOSUB 8010:REM DISPLAY FRACTION
  400 PRINT@392,"PLAY AGAIN <Y/N>?";
  410 A$=INKEY$:IF A$="" THEN 410
```

```
420 IF A$="Y" THEN 200
430 IF A$<>"N" THEN 400
440 CLS:END
3000 REM 2 SLICES OF PIZZA
3010 FOR L1=2 TO 34 STEP 2
3020 SET(L1,10,3)
3030 NEXT L1
3040 RETURN
3500 REM 3 SLICES OF PIZZA
3510 FOR L1=2 TO 18 STEP 2
3520 SET(L1,10,3)
3530 NEXT L1
3540 FOR L1=1 TO 7
3550 SET(18+L1,10+L1,3)
3560 SET(18+L1,10-L1,3)
3570 NEXT L1
3580 RETURN
4000 REM 4 SLICES OF PIZZA
4010 GOSUB 3010
4020 FOR L1=2 TO 18 STEP 2
4030 SET(18,L1,3)
4040 NEXT L1
4050 RETURN
4500 REM ** 5 SLICES
4510 FOR L1=2 TO 8 STEP 2
4520 SET(18,L1,3)
4530 NEXT L1
4540 SET(30,7,3):SET(27,8,3)
4550 SET(24,9,3):SET(21,10,3)
4560 SET(7,7,3):SET(10,8,3)
4570 SET(13,9,3):SET(16,10,3)
4590 FOR L1=2 TO 10 STEP 2
4600 SET(19-L1,L1/2+10,3)
4610 SET(17+L1,L1/2+10,3)
4620 NEXT L1
4640 RETURN
5000 REM *** 6 SLICES
5010 GOSUB 3500
5020 FOR L1=1 TO 7
5030 SET(18-L1,10+L1,3)
5040 SET(10+L1,2+L1,3)
5050 NEXT L1
5060 FOR L1=2 TO 16 STEP 2
5070 SET(18+L1,10,3)
```

```
5080 NEXT L1
5490 RETURN
5500 REM ** 7 SLICES OF PIZZA
5510 GOSUB 3000
5520 FOR L1=2 TO 8 STEP 2
5530 SET(18,L1+11,3)
5540 NEXT L1
5550 FOR L1=2 TO 8
5560 SET(9+L1,1+L1,3)
5570 SET(17+L1,11-L1,3)
5580 NEXT L1
5590 FOR L1=2 TO 10 STEP 2
5600 SET(19-L1,L1/2+10,3)
5610 SET(17+L1,L1/2+10,3)
5620 NEXT L1
5630 RETURN
6000 REM *** 8 SLICES
6010 GOSUB 4010
6020 FOR L1=2 TO 8 STEP 2
6030 SET(7+L1,L1/2+4,3)
6040 SET(30-L1,L1/2+4,3)
6050 SET(17-L1,L1/2+12,3)
6060 SET(20+L1,L1/2+12,3)
6070 NEXT L1
6080 RETURN
7000 REM FIGURE SUBROUTINE
7005 CLS0
7010 CO=240:REM ORANGE
7050 FOR P1=1 TO M: REM PEOPLE
7060 P=P(P1)
7070 FOR P3 = 1 TO 12: REM GRAPHICS
7080 PRINT@P+RC,CHR$((G(P3))+CO);
7090 RC=RC+1:IF RC=4 THEN RC=0:P=P+32
7100 NEXT:NEXT
7150 RETURN
8000 REM *** DISPLAY FRACTION
8010 CLS
8017 IF N/M=1 THEN PRINT@38,N/M;"SLICE P
ER PERSON":GOTO 8030
8020 PRINT@38,N/M;"SLICES PER PERSON"
8030 PRINT@101,N;"TOTAL PIZZA SLICES"
8060 PRINT@168,"EACH PERSON EATS"
8090 PRINT@204,N/M;"/";N
8100 PRINT@233,"OF THE PIZZA"
```

```
8110 FOR DELAY=1 TO 3000:NEXT
8120 RETURN
9000 REM *** DRAW PIZZA
9010 CO=4
9050 FOR L1=1 TO 13:SET(11+L1,0,CO):SET(
11+L1,21,CO):NEXT
9060 FOR L1=1 TO 6:SET(0,7+L1,CO):SET(36
,7+L1,CO):NEXT
9064 HP=13:VP=0:CL=1:FOR L1=1 TO 7
9066 SET(VP+CL+24,0+L1,CO)
9068 SET(VP+CL+24,21-L1,CO)
9071 SET(VP+CL,8-L1,CO)
9072 SET(VP+CL,HP+L1,CO):VP=VP+CL
9074 CL=CL+1:IF CL=3 THEN CL=1
9076 NEXT
9290 RETURN
9300 DATA 20,25,148,153,276,281,404,409
9310 DATA 1,13,14,2,3,11,7,3,1,14,13,2
```

Typing Hints ...

Several program lines contain words in lower case in the listing. This means they are to be typed in reverse video. In order to print the words in reverse video, press the **SHIFT** and **0** keys simultaneously. To turn off reverse video, press the **SHIFT** and **0** keys again.

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Background ...

The Pepperoni, Please! game was designed by Howard Boggess, a student at Patrick Henry High School in Roanoke, Virginia.

Howard had been using computers for five years. In 1978 his dad bought him an INTERACT computer. Although the computer cost $500, it was primitive compared to today's computers. It didn't understand decimal numbers or **strings** (letters and words). Still, for Howard, it was an excellent learning tool.

Howard began working on this game, Pepperoni, Please!, two weeks before it was due. But he was plagued with bad luck. He writes: "I took a computer home over a weekend, with only four days left to finish. I had finished six out of seven possible pizzas by Sunday morning at 3 AM. I was watching the snow fall outside my bedroom window and putting the finishing touch on the seventh pizza. That's when all the power went out and I lost everything!"

Howard worked all day Sunday on his program. He went to school Monday, but he got out of some of his classes to continue working on the program. He continues: "The program was due on Tuesday. Monday night I took the computer home and pieced together the whole program in three hours, eating pizza the whole time. After the program was finished I thought I'd be too sick to eat pizza ever again, but that's what I had for dinner that night."

Highlights ...

The computer draws the pizza and the people by using subroutines.

The subroutine beginning on line 9000 draws the pizza pie. The subroutine beginning on line 7000 draws the people.

The seven subroutines to draw the different number of pizza slices (randomly selected, from 2 to 8) begin on lines 3000, 3500, 4000, 4500, 5000, 5500, and 6000.

The subroutine to display the answer and the fraction of pizza allotted to each person begins on line 8000.

The computer chooses the number of pizza slices using an RND function on line 200.

Lines 220 to 325 take care of matching the number of randomly chosen pizza slices with the right number of people. To simplify the exercise, the computer chooses a number that results in each person getting whole slices of pizza (no fractional slices or pieces left over).

Variables . . .

A$	Your answer to the question "PLAY AGAIN?"
DELAY	Delay loop counter.
M	Number of slices in the pizza.
N	Number of people drawn on the screen.
YY	Your answer—number of slices per person.
YZ	Your answer—number of slices in the whole pizza.
P	Array—stores screen locations fro drawing people.
G	Array—stores graphics values for drawing people.
L1	Loop counter—used in drawing pizza and correct number of slices.
ERROR	If there are three incorrect answers (ERROR = 3) then the fraction is displayed.
RC	The amount added to the position of each person's figure so that the next point can be plotted.

Do-It-Yourself . . .

You might experiment with making the pizza pie and the people appear in different colors. You can do this by changing the color value on the SET commands before drawing either the pizza or the people.

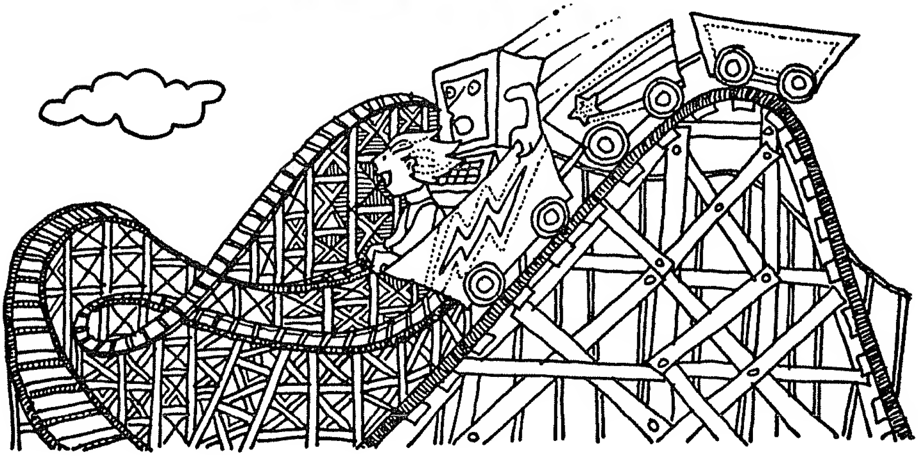
You can also add sound-effects, such as noises that sound like people munching and slurping juicy pizza pies.

Also, you can show how some of the fractions can be reduced. Sometimes the program (see the subroutine beginning on line 8000) prints a fraction like 2/4. It would be nice if the program showed that the fraction is equivalent to 1/2.

Finally, when the child answers the question correctly, you can have the number of pizza slices that go to one person turn a different color.

23

THE ROLLER COASTER



For Parents and Teachers ...

This game helps children learn about the sine function in trigonometry. It draws colorful sine curves on the TV screen. With modifications, the program can draw curves of the other trigonometric functions.

For Kids ...

How would you like to become a video artist? You can make interesting, beautiful pictures by using a few simple recipes. The recipes you use are mathematical formulas.

If you didn't use the formulas, you would have to tell the computer to draw each square or point in the picture. The formulas take care of calculating all of the points automatically.

Also, video art is a good way to spice up your math homework. By themselves, mathematical formulas can be complicated and boring. But they all describe some sort of shape. That shape might be beautiful. You can get your computer to draw the shapes on the TV screen. It makes math more interesting, and it is a lot easier than drawing the shapes yourself on a piece of graph paper.

This game is an example of video art using a sine function. The program draws a picture of a multi-colored roller coaster.

The Game ...

Program Name: **ROLLER**

```
50 REM *** ROLLER COASTER
62 CLS(0)
65 PRINT@260,"***   ROLLER COASTER!   **
*";
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 PI=3.14159
100 CLS(0)
130 FOR X=0 TO 63
132 DEG=360/63*X
135 RAD=DEG*PI/180
140 Y2=SIN(RAD)
145 Y=INT(-15*Y2+15)
150 SET(X,Y,RND(7))
152 FOR J=Y+2 TO 31
153 SET(X,J,7)
154 NEXT J
190 NEXT X
200 GOTO200
```

Highlights ...

On line 62, the screen is cleared and set to black. The value of π (3.14159) is defined on line 90. The program uses two loops to draw the curve of the roller coaster. The main loop—from lines 130 to 190—calculates the value of the point to be plotted and plots this point.

The X counter goes from 0 to 63, the horizontal SET values for the TRS-80 Color Computer screen. Line 132 changes these values (0 to 63) to degrees (0 to 360). Line 135 changes degree values to radians, since the arguments for the sine function must be in radians. The value of the sine function (Y2) is computed in line 140, a value btween −1 and 1. This value is then converted to an allowable vertical SET value (0 to 30) in line 145.

Finally, the point is plotted on line 150 in a random color. The J loop (lines 152 to 154) fills in all the points in a vertical bar below the sine point.

The program continues plotting points and drawing vertical bars until it reaches the right-hand edge of the screen.

Variables ...

PAUSE	Delay loop counter.
X	Loop counter—the horizontal SET value of the sine point.
PI	Constant—the value of π.
DEG	The X value changed to degrees.
RAD	The DEG value changed to radians.
Y2	The sine function value of RAD.
Y	The Y2 value changed to a vertical SET value.
J	Loop counter—the vertical point values below the sine SET point.

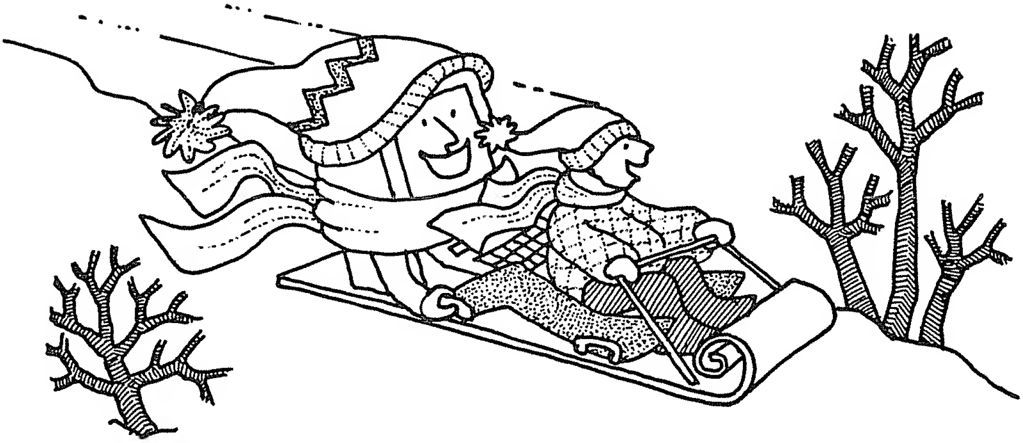
Do-It-Yourself ...

This program does more than produce video art. It enables you to visualize simple wave functions in a different way. It changes numbers and formulas into a beautiful picture. The program uses the SIN (sine) trigonometric function.

You can add a SOUND command to have the computer make a new sound as it draws each new line in the roller coaster—higher sounds for roller coaster peaks and lower sounds for dips. This will heighten the illusion that the computer (or child) is riding the roller coaster.

24

SLEDS!



For Parents and Teachers ...

This game is similar to The Roller Coaster game. It helps children learn about trigonometric functions. The computer draws one function—the cosecant function—on the TV screen. Children can experiment with the game and make all sorts of different shapes.

For Kids ...

In the last chapter we created a roller coaster. In this chapter we look at just how you go about using a derived function to create new pictures. By making only a couple of modifications to the roller coaster program, we end up with sleds!

The Game ...

Program Name: **SLEDS**

```
50 REM *** SLEDS
62 CLS(0)
65 PRINT@263,"***   SLEDS!   ***";
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 PI=3.14159
100 CLS(0)
130 FOR X=0 TO 63
132 DEG=360/63*X
135 RAD=DEG*PI/180
140 Y2=SIN(RAD)
143 IF Y2=0 THEN 190
144 CSC=1/Y2
145 IF ABS(CSC)>5 THEN 190
146 Y=-3*CSC+15
150 SET(X,Y,RND(7))
152 FOR J=Y+2 TO 31
153 SET(X,J,7)
154 NEXT J
190 NEXT X
200 GOTO200
```

Highlights ...

This program is similar to the ROLLER program. For details take a look at the last chapter. For this example I chose the cosecant function. To get the cosecant, you invert the sine.

CSC(X) = 1/SIN(X)

This is simple—except for one problem. We are using degrees (see the variable DEG on line 132), and when we compute the sine of 0 and 180 degrees, the answer is 0. We get the cosecant function above by dividing 1 by the sine of an angle. If the sine is 0 and we try to divide 1 by 0, we get an undefined number and the program crashes.

To solve this problem, we test for a sine equal to 0 on line 143. If the sine is equal to 0 we skip that value and proceed to the NEXT X statement (line 190).

The value of the cosecant is found on line 144 and is stored in the variable CSC. We then test to see if the absolute value of CSC is greater than 5 (line 145). If we don't, the values of CSC get too large to plot. Line 146 changes the allowable CSC values (−5 to +5) to allowable vertical SET values (0 to 30).

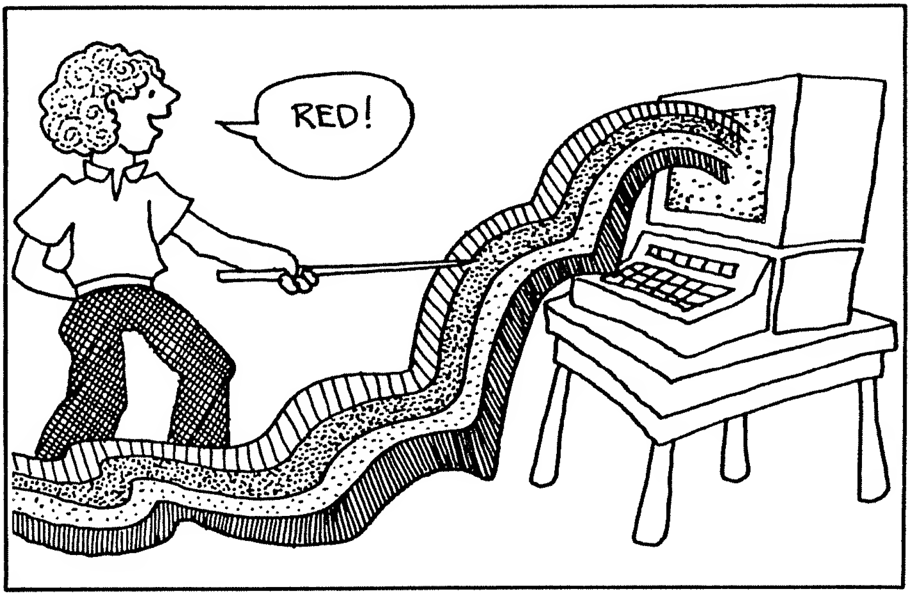
Finally, the point is SET on line 150 and the J loop (lines 152 to 154) fills in all the points in a vertical bar below the cosecant point.

Now the SLEDS program runs.

Do-It-Yourself ...

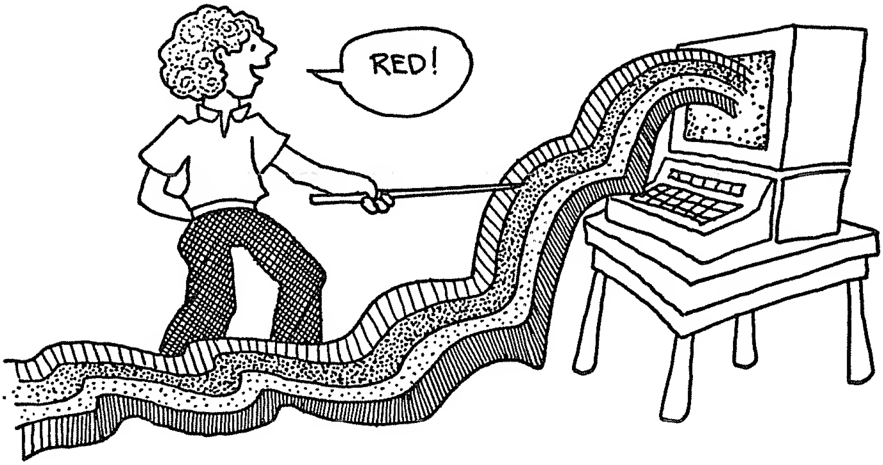
We've seen in this chapter that deriving a new trigonometric function can be tricky. But it's worth it. You can gain a lot of insight into the various trigonometric relationships. And, best of all, you can end up with some beautiful computer pictures.

COLORS



25

NAME THAT COLOR!



For Teachers and Parents ...

This game helps children learn colors and the color names. The computer flashes a color on the picture screen. It also displays a color name on the middle of the screen. If the color name does not match the color, the child presses the **SPACE** bar and a new color name appears. The child asks the computer for new color names until he finds the color name that matches the color. Then he presses the **ENTER** button.

For Kids ...

Pretend that you are in a carnival funhouse. Each of the rooms in the funhouse is a different color. You are stuck in a room until you guess its color. Then you can go on to the next room. Can you escape from the funhouse?

The Game . . .

Program Name: **COLOR**

```
50 REM *** NAME THAT COLOR
60 CLS
62 PRINT@35,"***  NAME THAT COLOR!  ***"
64 FOR PAUSE=1 TO 1000:NEXT PAUSE
69 DIM CHART$(9)
70 FOR L=1 TO 9:READ CHART$(L):NEXT L
72 FOR L=1 TO 9:READ CO(L):NEXT L
100 FOR I=1 TO 9
110 CLS CO(I)
130 J=1
135 PRINT@237,"       ";
140 PRINT@237,CHART$(J);
150 GUESS$="":NAME$=" "
160 K$=INKEY$:IF K$=""THEN 160
165 KEY=ASC(K$)
170 IF KEY=32 THEN 210
180 IF KEY<>13 THEN GOTO 160
185 GOSUB 2510
190 IF GUESS$=NAME$ THEN GOSUB 1010:GOTO
 220
200 GOSUB 2010
205 CLS CO(I):GOTO 135
210 J=J+1:IF J<10 THEN 135
215 IF GUESS$<>NAME$ THEN 130
220 NEXT I
250 CLS:END
2500 REM *** TEST ANSWER
2510 GUESS$=CHART$(J)
2520 NAME$=CHART$(I)
2560 RETURN
3000 REM COLOR DATA
3005 DATA BLACK,GREEN,YELLOW,BLUE,RED,BU
FF,CYAN,MAGENTA,ORANGE
3010 DATA 0,1,2,3,4,5,6,7,8
```

Highlights . . .

Be sure to LOAD in the Happy and Sad routine (FACES) before typing in this program.

The success of this program depends on how well the colors on the screen resemble the true colors. For example, if the computer's yellow looks more like orange, it is going to be hard for your child to get the right answer.

Try the program out on your TV set. To get the colors just right, you might have to play with the color and tint knobs.

This program is made of two loops and a new subroutine. The outer loop (lines 100 to 220) takes you through the 9 colored rooms in the funhouse: the Green Room, the Blue Room, the Buff Room, the Magenta Room, the Orange Room, the Red Room, the Cyan Room, the Black Room, and the Yellow Room.

The inner loop (lines 130 to 210) prints the names of the 9 colors, one at a time, on the TV screen (the wall of the room). If you think the name matches the color of the room, you press the **ENTER** button. If not, you press the **SPACE** bar, and a new name appears.

If you guess right, line 190 calls the Happy Face subroutine (GOSUB 1010). Then you leave the inner loop and go on to the next colored room in the funhouse.

If you guess wrong, line 200 calls the Sad Face subroutine (GOSUB 2010). Then the inner loop prints a new color name on the wall of the room. If the loop is complete and all the colors have been printed, then (on line 215) the program goes back to line 130, and the inner loop begins naming the colors again. You then have another chance.

When you have made it through all the rooms, the outer loop ends, and (on line 250) the program ends. You have escaped from the funhouse!

The new subroutine (lines 2500 and 2560) gets a color name stored in the color-name string array called CHART$ and stores the name in GUESS$. The color names flashed on the TV screen come from the array CHART$. The real color (to match the color on the screen) is switched into NAME$ on line 2520.

So before a comparison between NAME$ and GUESS$ can be made, the printed color is loaded into GUESS$.

After the color is in GUESS$ and NAME$, the subroutine ends, and (on line 190) the program looks for a match between GUESS$ (your guess) and NAME$ (the true color of the room).

Variables ...

PAUSE	FOR-NEXT loop counter—delay loop.
NAME$	The name of the color of a funhouse room.
GUESS$	The color name that you guess.
CHART$	String array containing all the possible colors.
I	FOR-NEXT loop counter (outer loop).
J	Loop counter (inner loop).
KEY$	The button you press—either the **ENTER** button (ASCII code = 13) to select a color, or the **SPACE** bar (ASCII code = 32) to have the computer print out a new color name.

Do-It-Yourself ...

Like the other programs in this book, Color can form the nucleus or core of a much more elaborate progam. As it stands, the program makes no reference to a funhouse. You can add PRINT commands to the beginning of the program and announce "WELCOME TO THE FUNHOUSE!" The PRINT commands can give children directions for escaping from the funhouse. You can even add sound and graphics commands to make the funhouse seem more realistic.

26

THE PIED PIPER

For Parents ...

This is a memory and concentration game. The computer creates a chain of colors and musical tones. The colors and tones are represented by the keys 1 to 4 on the computer keyboard. First the colors are displayed and the tones play, and then the child presses the keys. The keys have to be in the same sequence as the colors and tones. If the child can remember the exact sequence of colors and tones, the chain grows one link at a time. You can adjust how many colors and tones the child has to remember to win the game.

For Kids ...

In this game, the computer becomes a Pied Piper. It plays a musical tone and flashes a color on the TV screen, and you have to follow it.

The computer starts by playing a single tone and flashing a single color. If you press the right button, the same tone plays and the same color flashes on the TV screen. You won the game.

The computer asks you: "AGAIN? (PRESS ENTER)." If you press any other key, the game is over. If you press **ENTER**, the computer plays a new tone and flashes a new color. If you press the right button and match the color and tone, the computer then plays a second tone and a second color. Then it repeats the first tone and the first color. Now you have to press two buttons. The first button is for the first color and tone. The second button is for the second color and tone. If you get them both right, you win again.

But the computer still isn't done. Now it is ready to challenge you with three colors and three tones. If you press the right three buttons, then it comes back with four colors and tones—and on, and on.

How high does the computer go? The computer will play 50 games in a row. It has only 4 colors and 4 tones to work with. But it can piece these together into a string of up to 50 colors and tones. If you can remember 50 colors and tones in a row, the computer gives up. You are a genius!

The object of the game is to see how long you can follow the computer Pied Piper. It keeps stringing colors and tones together. And you try to match those colors and tones by pressing the color/tone buttons in the right order.

When the computer flashes the color red on the screen and plays a low F note, you press the **1** button.

When the computer flashes the color green and plays a middle F note, you press the **2** button.

When the computer flashes the color blue and plays a high F note, you press the **3** button.

And when the computer flashes yellow and plays a high-high F, you press the **4** button.

What happens if the computer strings four colors and tones in a row? Here's an example. If the computer:

FLASHES	RED	RED	YELLOW	GREEN
PLAYS	Low F	Low F	Hi-Hi F	Middle F

You wait until the computer plays the four colors and notes. Then you press four buttons: **1 - 1 - 4 - 2**. The computer will print a happy face. You won!

If you missed one of the colors (say you typed **1 - 1 - 3 - 2**), the computer will print a sad face and then ask you if you want to start a new game.

The Game ...

Program Name: **PIPER**

```
50 REM *** PIED PIPER
60 CLS
62 PRINT@35,"***   THE PIED PIPER   ***"
63 FOR PAUSE=1 TO 2000:NEXT PAUSE
64 CLEAR 500
65 DIM KEY(50)
70 DIM COMBO(50)
80 HUE(1)=4:HUE(2)=1:HUE(3)=3:HUE(4)=2:
90 PICH(1)=5:PICH(2)=133:PICH(3)=197:PIC
H(4)=229
95 MAX=1
100 OTAL=MAX
105 CLS0
107 GOSUB 800
110 COMBO(OTAL)=RND(4)
120 FOR PLAY=1 TO OTAL
130 N=PLAY:GOSUB 600:REM * PLAY/DISPLAY
132 FOR T=1 TO 800:NEXT T
135 NEXT PLAY
137 CLS0
140 PRINT@491,"YOUR TURN";
142 GOSUB 800
145 FOR TRY=1 TO OTAL
160 A$=INKEY$:IF A$="" THEN 160
162 IF ASC(A$) <49 OR ASC(A$) >52 THEN 1
60
164 KEY(TRY)=VAL(A$)
175 IF KEY(TRY) <> COMBO(TRY) THEN CLS:
GOSUB 2010:GOTO 230
177 N=TRY:GOSUB 600
200 NEXT TRY
```

```
205 FOR PAUSE=1 TO 600:NEXT
220 CLS:GOSUB 1010
222 GOTO 262
230 REM * PLAY AGAIN
235 PRINT@485,"AGAIN? (PRESS ENTER)";
237 A$=INKEY$:IF A$="" THEN 237
240 IF ASC(A$)=13 THEN MAX=1:GOTO 100
260 CLS:END
262 IFMAX<=50 THEN MAX=MAX+1:GOTO 100
264 CLS:END
600 REM *** COLOR AND SOUND
602 PRINT@392,CHR$(191);"=1";:PRINT@396,
CHR$(143);"=2";
604 PRINT@400,CHR$(175);"=3";:PRINT@404,
CHR$(159);"=4";
610 P=N:S=-2
615 FORL=1 TO P
620 S=S+3
625 IF L=11 THEN S=S+66
630 IF L=21 THEN S=S+66
640 IF L=31 THEN S=S+66
650 IF L=41 THEN S=S+66
660 NEXT L
670 TN=COMBO(N):TS=HUE(TN)
672 C=128+16*(TS-1)+15
680 PRINT@S,CHR$(C);:PRINT@S+1,CHR$(C);
690 PRINT@S+32,CHR$(C);:PRINT@S+33,CHR$(
C);
700 TN=COMBO(N):TS=PICH(TN)
710 SOUND TS,8
790 RETURN
800 REM *** COLOR CODE
810 PRINT@392,CHR$(191);"=1";:PRINT@396,
CHR$(143);"=2";
820 PRINT@400,CHR$(175);"=3";:PRINT@404,
CHR$(159);"=4";
830 RETURN
```

Typing Hints . . .

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program.

Highlights . . .

This program runs almost completely inside of a single FOR-NEXT loop (from lines 145 to 200). The loop controls the number of colors and tones the computer will string in a row. In the first game the computer challenges you with a single color and tone (MAX = 1). From then on, in each new game the computer adds a new color and tone (on line 262, MAX = MAX + 1).

Within the major loop is a second loop (on lines 120 to 135). This loop displays the colors and plays the tones in a single round of a single game.

Then comes a third loop, also inside the main loop (on lines 145 to 200). This loop lets you select the colors and tones for a single round of a single game.

If you push the right button (**1**, **2**, **3**, **4**) to select the right color and tone, the computer will display the color and play the tone. It then waits for you to input the next color and tone. As long as you keep entering the correct number, it continues until all colors and tones for that round are entered. If you select wrong, the computer prompts you for a new game. Press **ENTER** to start a new game, or any other key to quit.

Variables . . .

KEY — Array variable—used to translate the internal keyboard code to the numbers **1**, **2**, **3**, and **4**.

COMBO — Array variable—stores the number of the correct key (**1**, **2**, **3**, or **4**) for up to 50 rounds of colors and tones. The key number is selected at random (see line 110).

HUE — Array variable—contains the color value associated with each key.

PICH — Array variable—contains the tone value associated with each key.

OTAL	Counter—the number of colors and tones in the current round of the current game.
MAX	Counter—the total number of colors and tones in the current game.
PLAY	FOR-NEXT loop counter—the number of colors displayed and tones played by the computer in the current round of the current game.
T	FOR-NEXT loop counter—delay loop to play the computer's tone and display the computer's picture for a certain length of time.
TRY	FOR-NEXT loop counter—the number of colors and tones you have to select in the current round of the current game.
TS	Saves the color value from the array PICH.
TN	Saves the tone value from the array COMBO.
S	Current screen position.
L	FOR-NEXT loop counter—sets the value of S.
P	Last print position.
N	Next color value.
A$	Contains the internal keyboard code of the number button you press.
PAUSE	FOR-NEXT loop counter—delay loop (line 205).

Do-It-Yourself ...

You might want to change the MAX variable to something less miraculous than 50. For example, if you have a five-year-old playing the Pied Piper game, you might set MAX (see line 262) to go only up to 5 or 10. Then, if the child is able to follow the Pied Piper for 5 or 10 colors and tones, you can reward her. You can put some PRINT commands after line 262 congratulating her on doing so well.

Also, you can put some PRINT commands up front to introduce the Pied Piper and explain the rules of the game.

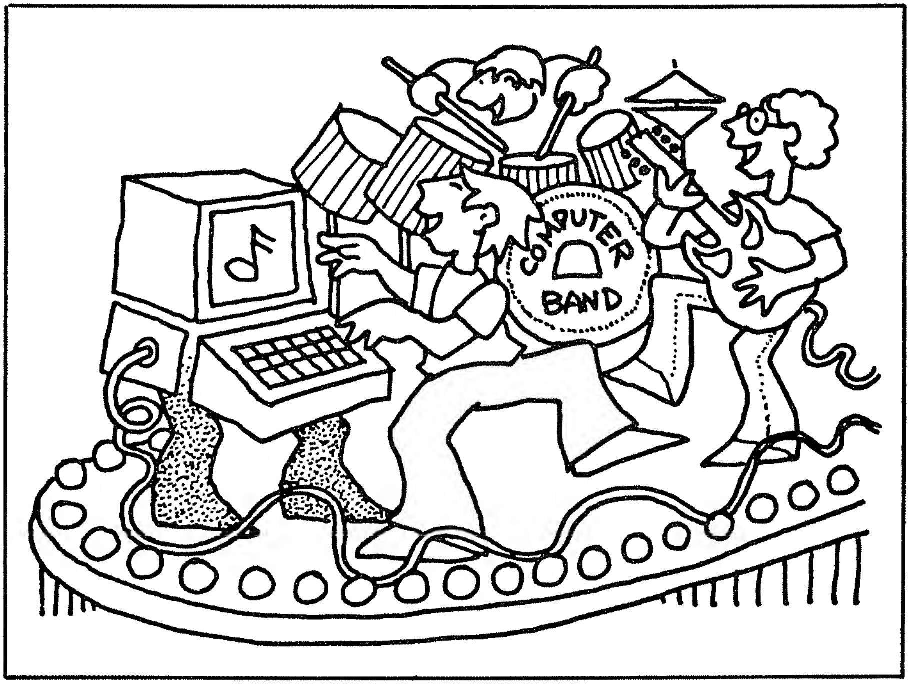
You can also make the game harder. You can create a Super Piper game by adding new tones and colors. Presently the game has only four colors and tones to work with. If you increase this number, a child will have to keep track of more sounds and colors.

You can also speed the game up. You can make the computer play the tones and display the colors in a flash by modifying the delay loop on line 133. For example, you can change the loop from FOR T=1 TO 800 to FOR T=1 TO 200.

If anyone ever beat the computer at the Super Piper game, it would be a true feat of mental wizardry.

Perhaps you and your children are up to the challenge!

MUSIC



27

DO-RE-MI

For Parents and Teachers . . .

This game helps children learn the names of the notes of the C-major scale in the octave from middle C to high C. It also helps them learn the pitch of each note.

For Kids . . .

Pretend you are sitting outside, on the grass, on a warm summer evening. You look up at the stars and the moon, and you hear music. Actually, what you hear is a single note floating somewhere, invisible, high in the air.

You have your flashlight with you. You turn on the light and take out your **Kid's Guide to Nighttime Music.** You try to identify the musical note playing in the dark sky.

Is it C? D? E? F? G? A? B? Or High C?

You guess that it is middle C. You look up at the moon and hum the note. “Middle C,” you say. “Is it Middle C?”

The moon begins to smile. “Right!” it calls. The moon and stars play a happy tune.

Now a new mystery note glides through the air. You switch your flashlight back on and turn your book’s pages. Is the new note a middle C? Is it a D? An E?

The Game ...

Program Name: **NOTEGAME**

```
50 REM *** DO-RE-MI GAME
60 CLS
65 PRINT@36,"***  DO-RE-MI  GAME  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM KEY(8)
90 DIM MUSIC(8)
92 GOSUB 3010
100 CLS
110 GOSUB 3510
120 FOR N=1 TO 8
130 SOUND MUSIC(N),32
140 J=1
142 PRINT@238,"    ";:REM 4 SPACES
145 IF J<>8 THEN 150
147 PRINT@238,"HI-";CHR$(KEY(J));
148 GOTO 170
150 PRINT@239,CHR$(KEY(J));
170 K$=INKEY$:IF K$="" THEN 170
172 K=ASC(K$)
180 IF K=32 THEN 230
190 IF K<>13 THEN 170
200 IF J=N THEN GOSUB 1010:GOTO 240
210 GOSUB 2010:SOUND MUSIC(N),24:GOSUB 3
510
230 J=J+1:IF J<9 THEN 142
235 GOTO 140
240 IF N<8 THEN GOSUB 3510
245 NEXT N
250 PRINT@448,"  ";
```

```
260 PRINT"PLAY AGAIN";:INPUT A$
270 IF LEFT$(A$,1)="Y" THEN GOTO 100
280 IF LEFT$(A$,1)<>"N" THEN 250
290 CLS:END
3000 REM *** LOAD KEYS
3010 FOR I=1 TO 8
3020 READ MUSIC(I)
3030 READ KEY(I)
3040 NEXT I
3050 RETURN
3500 REM *** SCREEN MESSAGE
3510 PRINT@132,"***   WHICH NOTE?   ***"
;
3520 RETURN
5000 DATA 89,67,108,68,125,69,133,70,147
,71,159,65,170,66,176,67
```

Highlights ...

Be sure to LOAD the Happy and Sad routine (FACES) before typing in this program.

When you RUN this program, the computer plays a single musical note. On the screen, in black on green, appears a note, and the computer prints the question: "WHICH NOTE?"

If you believe that the note on the screen matches the note played, press the ENTER button. If not, press the SPACE bar. Each time you press the SPACE bar, the note on the screen changes to the next higher note, until you reach high C. Then the computer goes back to the beginning of the octave and displays the first note, (middle) C.

If you get a note wrong, the computer makes a sad face. Then it keeps playing the same note until you guess it right.

If you get a note right, the computer makes a happy face. Then it tests you on the next note in the octave. The note displayed on the screen goes back to middle C. You have to press the SPACE bar in order to change the note to match the new note being played.

The computer lets you correctly guess eight notes. Then it asks you if you want to "PLAY AGAIN?" If you type "Y" or "YES," you get to play a new game with eight more notes.

Variables . . .

PAUSE Delay loop counter.

A$ Accepts answer to "PLAY AGAIN?" question.

KEY Array—button associated with each note—**C**, **D**, **E**, **F**, **G**, **A**, **B**, and **C** (for high C).

MUSIC Array—musical tones (middle C through high C).

N Loop counter—major loop (lets you guess eight notes).

J Loop counter—selects the note displayed on the TV screen.

K Your guess ("WHICH NOTE?").

I Loop counter—in subroutine (beginning on line 3000) for reading in eight musical notes and codes for the eight keys representing the musical notes **C**, **D**, **E**, etc.

Do-It-Yourself . . .

You can do well at this game by keeping track of the last note the computer played. The next note the computer plays will always be one note higher.

For a look at a game that scrambles the notes, turn to the next chapter.

28

SCRAMBLED NOTES

For Parents and Teachers . . .

This game will help children learn musical notes of the C-major scale in the octave from middle C to high C. Since the notes are randomly selected by the computer, the game helps children learn to identify the different notes just by hearing them.

For Kids . . .

Take a look at the For Kids section in the last chapter.

A hint: This game is tougher than the last game. In this game the notes don't get higher one by one. In this game, the notes are all scrambled. You never know which note will be played next!

The Game . . .

Program Name: **NOTEGAM2**

```
50 REM *** SCRAMBLED NOTES
60 CLS
65 PRINT@34,"***   SCRAMBLED NOTES   ***
"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM KEY(8)
90 DIM MUSIC(8)
92 GOSUB 3010
100 FIRST=1:LAST=8:HOP=1:GOSUB 4010
103 FOR PAUSE=1 TO 600:NEXT PAUSE
105 FIRST=8:LAST=1:HOP=-1:GOSUB 4010
106 FOR PAUSE=1 TO 600:NEXT PAUSE
110 GOSUB 3510
120 FOR O=1 TO 10
125 N=RND(8)
130 SOUND MUSIC(N),24
140 J=1
142 PRINT@238,"    ";:REM 4 SPACES
145 IF J<> 8 THEN 150
147 PRINT@238,"HI-";CHR$(KEY(J));
148 GOTO 170
150 PRINT@239,CHR$(KEY(J));
170 K$=INKEY$:IF K$="" THEN 170
174 K=ASC(K$)
180 IF K=32 THEN 230
190 IF K<>13 THEN 170
200 IF J=N THEN GOSUB 1010:GOTO 240
210 GOSUB 2010:GOSUB 3510:SOUND MUSIC(N)
,24
230 J=J+1:IF J<9 THEN 142
235 GOTO 140
240 IF O<10 THEN GOSUB 3510
245 NEXT O
260 CLS:PRINT@461,"AGAIN";:INPUT A$
270 IF LEFT$(A$,1)="Y" THEN GOTO 100
280 IF LEFT$(A$,1)<>"N" THEN 260
290 CLS:END
3000 REM *** LOAD KEYS
3010 FOR I=1 TO 8
3020 READ MUSIC(I)
```

```
3035 READ KEY(I)
3040 NEXT I
3050 RETURN
3500 REM *** SCREEN MESSAGE
3510 CLS 2
3512 PRINT@132,"***   WHICH NOTE?   ***"
;
3520 RETURN
4000 REM *** PLAY NOTES
4010 CLS 2
4012 PRINT@34,"***  LISTEN TO THE NOTES
 ***";
4015 FOR J=FIRST TO LAST STEP HOP
4040 PRINT@238,"    ";
4050 IF J<> 8 THEN 4080
4060 PRINT@238,"HI-";CHR$(KEY(J));
4070 GOTO 4090
4080 PRINT@239,CHR$(KEY(J));
4090 SOUND MUSIC(J),16
4092 FOR PAUSE=1 TO 400:NEXT PAUSE
4100 NEXT J
4110 RETURN
5000 DATA 89,67,108,68,125,69,133,70,147
,71,159,65,170,66,176,67
```

Highlights . . .

This game is very similar to the game in the last chapter, but there are some changes.

First, when the game begins, the computer plays all eight notes in the octave and displays the note names on the TV screen. Then it plays the notes backwards, beginning at high C and ending at middle C. While the computer is playing, listen carefully and try to associate each note name with the note being played.

Next the computer erases the screen and randomly picks a note from among the eight notes in the octave. This makes the game much more challenging than the previous game in which the computer always picked the next higher note.

Also, if you guess a note wrong, the computer doesn't go back and display the first note, middle C. This time it displays the note following the note you guessed (incorrectly).

In the last game, you had to guess eight notes correctly to complete the game. In this game, you have to guess 10 notes correctly.

Variables ...

N Stores the random note (any note from middle C to high C).

O Loop counter—major program loop (controls the number of notes you will be challenged with).

J Loop counter—in subroutine beginning at 4000 (controls playing all the notes in the octave).

FIRST First note to be played in the loop.

LAST Last note played in the loop.

HOP Increment by which the loop is increased or decreased. The first time around, HOP is +1, so the computer plays the octave forward from middle C to high C. The second time, HOP is -1, so the computer plays the octave **backwards** from high C to middle C.

KEY Array—button associated with each note—C,D,E,F,G,A,B, and C (for high C).

MUSIC Array—musical tones (middle C through high C).

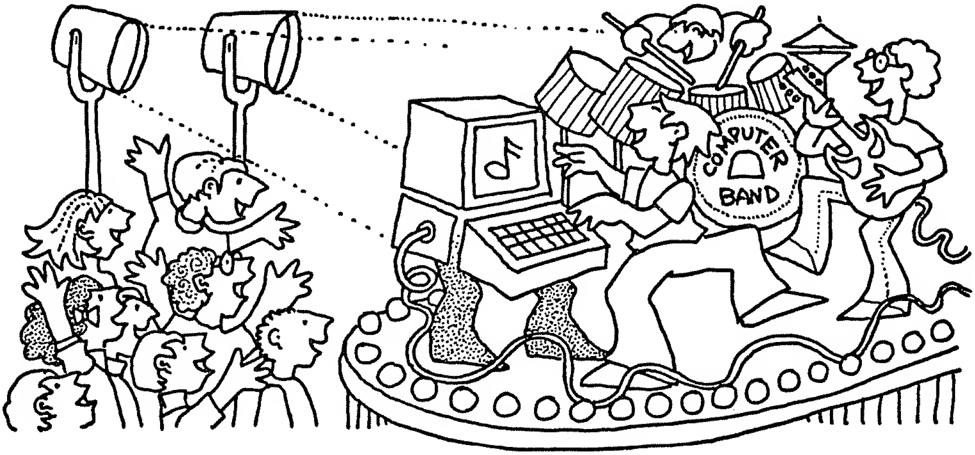
Do-It-Yourself ...

This game plays only the notes of the C-major scale in the octave from middle C to high C. It would be neat if you added the other major and minor scales and expanded the number of notes to include lower (bass) octaves and higher (treble) octaves. This would make the game more challenging and would help children learn more about music.

Also, the game, as it now stands, only uses notes. You can include other facets of a musical score, such as rests and tempo. The game can play a little tune using different tempos (4/4 time, 3/4 time, etc.), and the children can try to guess which. Also, you can have the computer play a scale, and the children can try to guess its name.

29

MAKE UP A SONG



For Parents and Teachers ...

This game will help children learn how to create musical notes and tunes. It will teach them how different notes can be combined in beautiful, harmonious, or unusual ways.

For Kids ...

Pretend that your computer is an electronic guitar. You pick notes on a guitar. You combine the notes into songs. Now you can do that on your computer, too.

When you RUN the Notes program, it will first play all the notes between middle C and high C (including the sharps). Listen carefully to these notes as the computer plays them.

Then the computer will ask you to "PICK A NOTE." To choose the notes in your tune, you need to press only one of two buttons—the **SPACE** bar or the **ENTER** button. You press the

ENTER button to select a note to go in your tune. You press the **SPACE** bar to change the notes on the TV screen until the note displayed is the one you want to select.

You can create a song out of notes—one note played right after the other. This game will let you create a song with up to 10 notes.

After you have created a single tune, the computer will ask you if you want to create a new tune. Answer "Y" to create a new one. If you answer "N," the computer will play the notes in your existing song.

After the computer has played all the notes, it will ask you if you want to hear the old tune again. If you type "N," it will ask you if you want to make a new tune. If you don't, the game will end. But if you do want to make a new tune and answer "Y," the computer will erase your old song and let you create a new one.

The Game ...

Program Name: **NOTES**

```
50 REM *** PLAY TUNES GAME
55 CLS
60 PRINT@34,"***   PLAY-A-TUNE-GAME   **
*"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DU=6:REM * DURATION OF NOTE
75 DIM PY(10)
80 DIM KEY(13), SHARP(13)
90 DIM MUSIC(13)
92 GOSUB 3010
100 FIRST=1:LAST=13:HOP=1:GOSUB 4010
103 FOR PAUSE=1 TO 600:NEXT PAUSE
105 FIRST=13:LAST=1:HOP=-1:GOSUB4010
106 FOR PAUSE=1 TO 600:NEXT PAUSE
117 C=1
119 GOSUB 3510
140 FOR J=1 TO 13
142 PRINT@224," "
145 IF J <> 13 THEN 150
147 PRINT@239,"HI-";CHR$(KEY(J))
```

```
148 GOTO 160
150 PRINT@239,CHR$(KEY(J));CHR$(SHARP(J)
)
160 SOUND MUSIC(J),DU
162 IF C=11 THEN 230
170 A$=INKEY$:IF A$=""THEN 170
172 K=ASC(A$)
180 IFK=32 THEN 230
190 IF K <> 13 THEN 170
200 PY(C)=J
210 C=C+1
230 NEXT J
232 PRINT@420," ";
234 PRINT"1= PLAY TUNE  2=ADD NOTES";
236 A$=INKEY$:IFA$=""THEN 236
238 IF VAL(A$) <1 OR VAL(A$)>2 THEN 236
240 IF VAL(A$)=1 THEN C=C-1:GOTO300
250 PRINT@420," ":GOTO 140
300 GOSUB 4510
310 CLS
320 PRINT@456,"PLAY TUNE AGAIN?";
330 A$=INKEY$:IFA$=""THEN 330
335 IF A$="Y" THEN 300
337 IF A$ <> "N" THEN 320
340 PRINT@456," "
343 PRINT@456,"PLAY A NEW TUNE?";
345 A$=INKEY$:IF A$=""THEN 345
347 IF A$="Y" THEN CLS:GOTO 117
349 IF A$<>"N" THEN 340
350 CLS:END
3000 REM *** LOAD KEYS
3010 FOR L=1TO13
3020 READ MUSIC(L):NEXT
3030 FOR L=1TO13:READ SHARP(L):NEXT
3040 FOR L=1TO13:READ KEY(L):NEXT
3050 RETURN
3500 REM *** SCREEN MESSAGE
3510 PRINT@99," ";
3515 PRINT"***   PICK A NOTE   ***"
3520 RETURN
4000 REM *** PLAY NOTES
4010 CLS
4012 PRINT@98,"*** LISTEN TO THE NOTES *
**"
4015 FOR J=FIRST TO LAST STEP HOP
```

```
4040 PRINT@236," "
4050 IF J<>13 THEN 4080
4060 PRINT@236,"HI-";CHR$(KEY(J));CHR$(S
HARP(J))
4070 GOTO 4090
4080 PRINT@236,CHR$(KEY(J));CHR$(SHARP(J
))
4090 SOUND MUSIC(J),DU
4092 FOR PAUSE=1 TO 200:NEXT PAUSE
4100 NEXT J
4105 PRINT@96," "
4110 RETURN
4500 REM *** PLAY NOTES IN TUNE
4510 CLS
4512 PRINT@98,"*** NOTES IN YOUR TUNE **
*"
4513 FOR F=1 TO C
4515 E=PY(F)
4550 IF E<>13 THEN 4580
4560 PRINT@236,"HI-";CHR$(KEY(E));CHR$(S
HARP(E))
4570 GOTO 4590
4580 PRINT@236,CHR$(KEY(E));CHR$(SHARP(E
))
4590 SOUND MUSIC(E),DU
4592 FOR PAUSE=1 TO 50:NEXT PAUSE
4595 NEXT F
4600 FOR PAUSE=1 TO 400:NEXT PAUSE
4605 CLS
4610 RETURN
5000 DATA 89,99,108,117,125,133,140,147,
153,159,165,170,176
5020 DATA 32,35,32,35,32,32,35,32,35,32,
35,32,32
5030 DATA 67,67,68,68,69,70,70,71,71,65,
65,66,67
```

Highlights ...

This program is similar to the other two music programs in the previous chapters—NOTEGAME and NOTEGAM2. It shows how you can take simple programs and use them as building blocks to make a program that is more elaborate and complex.

Look at the DATA commands on lines 5000 to 5030. The first 13 values are for the musical tones stored in array MUSIC. The next 13 values are for the ASCII codes for a space or a "#" symbol stored in array SHARP. If the value is not a "sharp" (an ASCII 35), it is a space (an ASCII 32). The last 13 values are codes for the buttons that select the note (a "C," a "D," an "E," etc.) that are stored in array KEY.

Each time you select a note, the value of the note (a pointer to the actual note values stored in the MUSIC array and the note-name values stored in the KEY array) is stored in an array called PY. There are 10 columns in the PY array to store the (up to) 10 notes you have selected for your song.

The new chief subroutine is the Play Notes in Tune subroutine beginning on line 4500. This subroutine looks at the PY array and selects the notes, one at a time. The Play Notes subroutine plays the individual notes in each tune.

Variables ...

PY Array—the values in PY act as pointers to the musical note values stored in the MUSIC array and the note-name values stored in the KEY array.

SHARP Array—stores either a space or a sharp value (#) to go with each musical note displayed on the TV screen.

KEY Array—stores 13 letter values representing the note (A through G).

C Loop counter—major loop (controls number of notes chosen—you may choose up to 10 notes).

J Loop counter—controls the 13 notes that you have to choose from (displays the notes on the TV screen) and, in subroutine beginning at line 4000, controls playing all the notes in the octave.

F Loop counter in Play Notes subroutine—controls playing of each of up to 10 notes you have selected.

FIRST First note to be played in the loop.

LAST Last note played in the loop.

HOP Increment by which the loop is increased or decreased. The first time around, HOP is +1, so the computer plays the octave forward from middle C to high C. The second time, HOP is -1, so the computer plays the octave **backwards** from high C to middle C.

DU Duration of a note.

E Loop counter in Play Notes subroutine—controls playing of the notes in each tune.

PAUSE Delay loop counter.

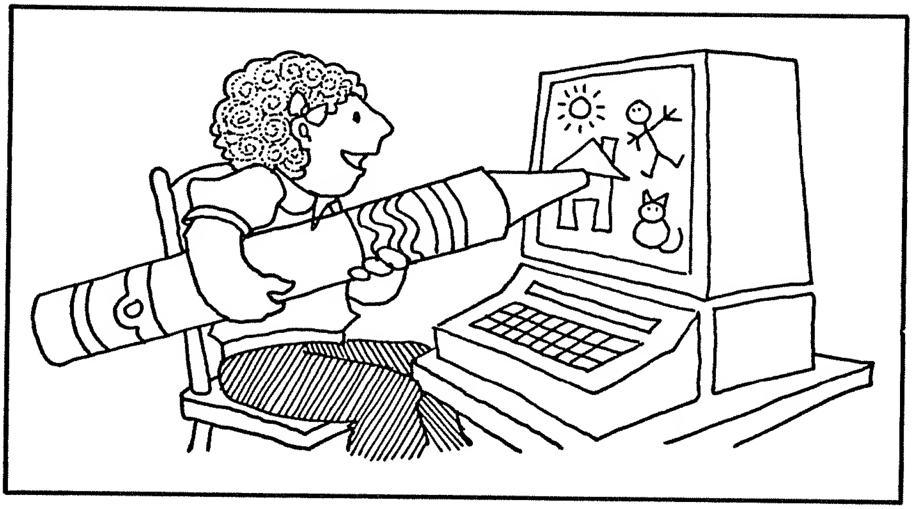
MUSIC Array—musical tones (middle C through high C).

Do-It-Yourself . . .

This program can be simplified or be made even more elaborate. For example, you might want to create a simpler game in which children compose a song out of fewer notes. Or you can load in a larger number of notes (see the table of Pitch Values for Musical Notes in the Appendix), so that children can create tunes from up to three octaves on the musical keyboard.

Finally, you might also consider giving children the chance to select the tempo of the song and other components of the musical score such as note duration. For example, children could select eighth-notes, quarter-notes, half-notes, or whole-notes.

DRAWING



30

THE COMPUTER CRAYON

For Parents and Teachers ...

This game turns the computer into an electronic crayon and the TV screen into a sketchpad. This game appeals to all ages. Yet even a younger child can use the game to develop his or her artistic abilities. And it's a great imagination game. The TV screen is empty, and the child can fill it with whatever he or she dreams up.

For Kids ...

Load the Crayon program and type RUN.

The Crayon color selection menu appears on the screen. It can be a yellow, green, blue, or red crayon, or it can be a special crayon that makes all four colors. When you have picked out the color of your crayon, type the crayon number (1, 2, 3, 4, or 5), the screen will darken (about the color of a chalkboard at school), and you can immediately begin drawing. The screen is an electronic sketchpad. You can draw anything you want.

The way you draw is by using the **arrow** buttons on the right side of the keyboard. To draw upward you press the **up-arrow** (↑) button. To draw downward you press the **down-arrow** (↓) button. To draw to the right you press the **right-arrow** (→) button. To draw to the left you press the **left-arrow** (←) button.

Try it out.

What happens when you come to the edge of the screen? Right. The computer stops you. You stay right there until you push a new arrow button and draw in a different direction.

Now that you know how to draw, you need to know something else: how to erase. Erasing is as easy as drawing. You decide which direction you want to erase, you press the correct arrow button, and at the same time, you hold down the **SHIFT** button (on either side of the keyboard).

Try erasing. Backtrack over some of the lines you just drew on your electronic paper. As long as you hold down the **SHIFT** button, the crayon doesn't draw anything. But you can still see it. It's like a ghost. It blinks off and on, so you know exactly where it is.

What if you have drawn a picture, and you want to draw something new? Do you have to erase each little color block on the TV screen?

No. Just press the **ENTER** button. Poof! Your drawing vanishes. Now you can try something new.

What should you draw? You can draw anything! I love to draw monster faces, giant letters, and railroad tracks. You might try some mazes, animals, bridges, and planets. Or some beautiful shapes and patterns—floating cubes, squares, spirals, and boxes inside boxes.

The screen is empty. Pick out a crayon and start drawing.

To SAVE the TV screen drawing, follow this procedure:

1. LOAD and RUN the Crayon program.
2. Draw a picture on the TV screen.

3. Press **S** when you are ready to SAVE the picture.
4. Decide on a name for the picture.
5. Select **T** for tape or **D** for disk when prompted.
6. Position tape.
7. Press PLAY and RECORD on the tape player.
8. Name your picture.
9. Press **ENTER** key when ready.

To LOAD a previously SAVEd TV screen drawing, follow this procedure:

1. LOAD and RUN the Crayon program.
2. Press **L** when you are ready to LOAD in a picture.
3. Remember the name for the previously SAVEd picture. Rewind the tape to the location (counter number) where the picture was SAVEd.
4. Press PLAY on the recorder.
5. Type picture name when prompted.
6. Press **ENTER** key when ready.

Your TV screen will then display the previously SAVEd picture. You can change the drawing, if you wish, and SAVE it again under a different name, or the same name.

The Game . . .

Program Name: **CRAYON**

```
50 REM *** COMPUTER CRAYON
52 CLEAR 1000
54 DIM PIC(512)
55 CLS
```

```
60 PRINT@34,"*** THE COMPUTER CRAYON ***
"
65 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM DRAW(4)
72 FOR J=1 TO 4
74 READ DRAW(J)
78 NEXT J
100 GOSUB 4010:REM * CHOOSE COLOR
102 CLS 0:REM BLACK
105 X=7:Y=15:REM CENTER
106 OX=X:OY=Y
110 T=0
120 IF C=5 THEN N=DRAW(RND(4)):GOTO 130
125 N=DRAW(C)
130 K$=INKEY$:IF K$="" THEN S=N:N=128:GO
TO 175
131 K=ASC(K$)
132 IF K=13 THEN 102
133 IF K=83 THEN 500
134 IF K=76 THEN 700
135 IF K=21 OR K=93 OR K=95 OR K=91 THEN
 S=N:N=128:OX=X:OY=Y
140 IF K=8 OR K=21 THEN Y=Y-1:GOTO 174
150 IF K=9 OR K=93 THEN Y=Y+1:GOTO 174
160 IF K=94 OR K=95 THEN X=X-1:GOTO 174
170 IF K=10 OR K=91 THEN X=X+1:GOTO 174
172 GOTO 130
174 GOSUB 3010:REM * ERROR CHECK
175 IF N=128 THEN PRINT@X*32+Y,CHR$(S);:
FOR U=1 TO 10:NEXT U:PRINT@X*32+Y,CHR$(N
);:GOTO 185
176 PRINT@OX*32+OY,CHR$(N);
185 OX=X:OY=Y:T=T+1:IF T<5 THEN 120
190 GOTO 110
500 REM * SAVE DISPLAY *
510 FOR LOC=1 TO 512
515 PIC(LOC)=PEEK(1023+LOC)
520 NEXT LOC
522 CLS
524 PRINT@32,"T=TAPE OR D=DISK";:INPUT S
D$
526 IF SD$ <> "T" AND SD$ <> "D" THEN 52
4
528 IF SD$="D" THEN 600
532 MOTORON
```

```
535 PRINT@32,"1. POSITION TAPE"
537 INPUT"   (PRESS <ENTER> WHEN READY";
R$
539 MOTOROFF
540 PRINT "2. PRESS PLAY AND RECORD"
545 INPUT"3. NAME YOUR PICTURE";FI$
550 PRINT:INPUT"4. PRESS <ENTER> KEY WHE
N READY";R$
560 OPEN "O",#-1,FI$
570 FOR LOC=1 TO 512
575 PRINT #-1,PIC(LOC)
577 POKE 1023+LOC,PIC(LOC)
580 NEXT LOC
585 CLOSE #-1
590 GOTO 130
600 PRINT:INPUT"NAME YOUR PICTURE";FI$
610 OPEN "O",#1,FI$
615 FOR LOC=1 TO 512
620 PRINT #1,PIC(LOC)
630 POKE 1023+LOC,PIC(LOC)
640 NEXT LOC
650 CLOSE
660 GOTO 130
700 REM * LOAD DISPLAY *
702 CLS
704 PRINT@32,"T=TAPE OR D=DISK";:INPUT S
D$
705 CLS
706 IF SD$ <> "T" AND SD$ <> "D" THEN 70
4
708 IF SD$="D" THEN 800
709 MOTORON
710 PRINT@32,"1. POSITION TAPE"
712 INPUT"   (PRESS <ENTER> WHEN READY)"
;R$
714 MOTOROFF
720 PRINT"2. PRESS PLAY ON RECORDER"
725 INPUT"3. ENTER PICTURE NAME";FI$
730 PRINT:INPUT"   PRESS <ENTER> WHEN RE
ADY";R$
740 OPEN "I",#-1,FI$
745 FOR LOC=1 TO 512
750 IF EOF (-1) THEN 790
755 INPUT #-1,PIC(LOC)
760 POKE 1023+LOC,PIC(LOC)
```

```
780 NEXT LOC
790 CLOSE #-1
795 GOTO 130
800 PRINT:INPUT"ENTER PICTURE NAME";FI$
810 OPEN "O",#1,FI$
815 FOR LOC=1 TO 512
820 IF EOF (#1) THEN 850
825 INPUT #1,PIC(LOC)
830 POKE 1023+LOC,PIC(LOC)
840 NEXT LOC
850 CLOSE
860 GOTO 130
3000 REM *** ERROR CHECK SUBROUTINE
3010 IF X<0 THEN X=0
3020 IF X>15 THEN X=15
3030 IF Y<0 THEN Y=0
3040 IF Y>31 THEN Y=31
3050 IF X=15 AND Y=31 THEN X=14:Y=30
3060 RETURN
4000 REM *** PICK COLOR
4010 CLS
4030 PRINT@98,"1.   RED"
4040 PRINT@130,"2.   GREEN"
4050 PRINT@162,"3.   BLUE"
4060 PRINT@194,"4.   YELLOW"
4065 PRINT@226,"5.   ALL COLORS"
4070 PRINT@258,"COLOR (1,2,3,4,5)";:INPU
T C$
4072 C=VAL(C$)
4075 IF C>=1AND C<=5 THEN 4080
4076 GOTO 4010
4080 RETURN
5000 DATA 191,143,175,159
```

Highlights ...

On lines 72 to 78, the program loads 4 numbers into the DRAW array. These numbers represent the color of the crayon. The 191 in graphics character code represents a red block. When the computer obeys a PRINT CHR$ (191), a red block appears.

Likewise, the 143 code represents a green block, and the 175 represents a blue block. A PRINT CHR$ (143) will make a green block; a PRINT CHR$ (175) will make a blue block.

You select the color using the subroutine beginning on line 4000.

All the drawing takes place inside the loop on lines 110 to 185. Elaborate checks are made for each key you press.

First, on line 132, the computer checks to see if you pressed the **ENTER** button. If so, it leaves the drawing loop, erases the screen, and restarts the drawing loop.

Second, on lines 133 and 134, the computer checks to see if you pressed the **S** button to save a drawing or the **L** button to load a drawing.

Third, the computer checks to see if you pressed the arrow keys alone or if you pressed the arrow keys along with the **SHIFT** button. If you pressed the **SHIFT** button, that means you wanted to erase instead of draw.

Fourth, the computer calls a subroutine (at line 3000) to make sure that you are not about to "fall off" the TV screen.

Line 175 takes care of erasing. It displays the drawing cursor for a count of 10. Then it erases the cursor and the current color block. It does the erasing with a PRINT CHR$(N), where N=128 (a black block).

The subroutines in lines 500 to 660 and 700 to 860 allow drawings to be loaded and/or saved to tape or disk and provide for extensive error checking.

Variables . . .

PAUSE	Delay loop counter.
DRAW	Array—holds screen color values.
J	Loop counter.
K	Reads in arrow value.
X	Row where color block is drawn.

Y	Column where color block is drawn.
T	Loop counter—main drawing loop.
N	Color value for current color block.
C	Points to current color in crayon array DRAW.
S	Indicates color value (of N) when you want to erase.
PIC	Array—stores screen picture when saving.
LOC	Loop counter—used for saving and loading pictures.

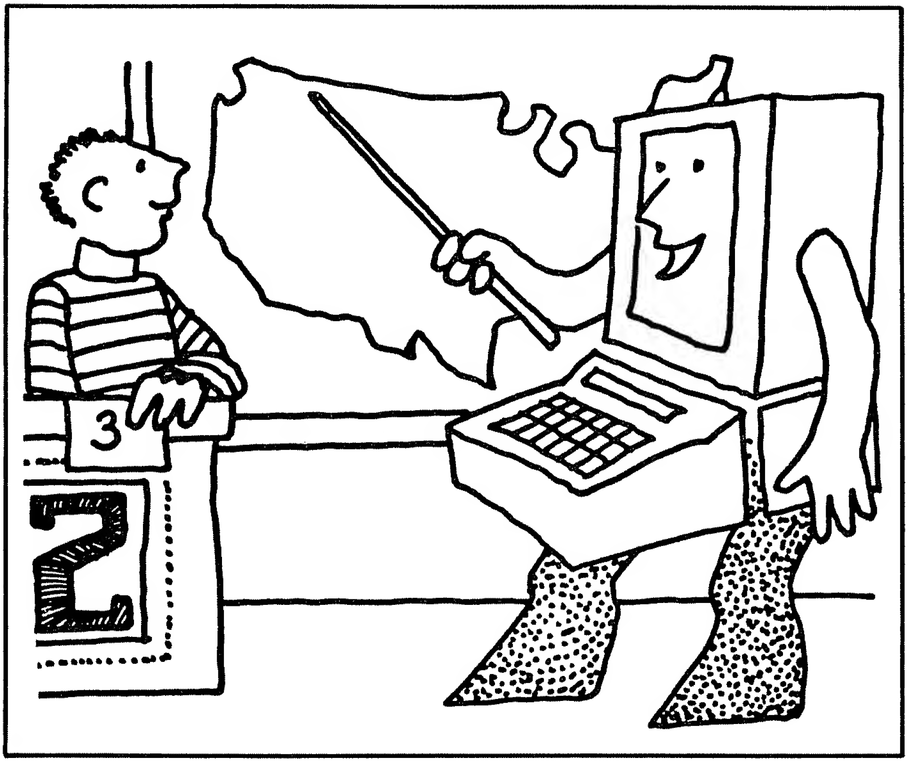
Do-It-Yourself . . .

There are dozens of things you can do to improve the Crayon program. For example, you can increase the number of crayon colors the children can choose from.

You can also modify the program to allow changing colors while drawing.

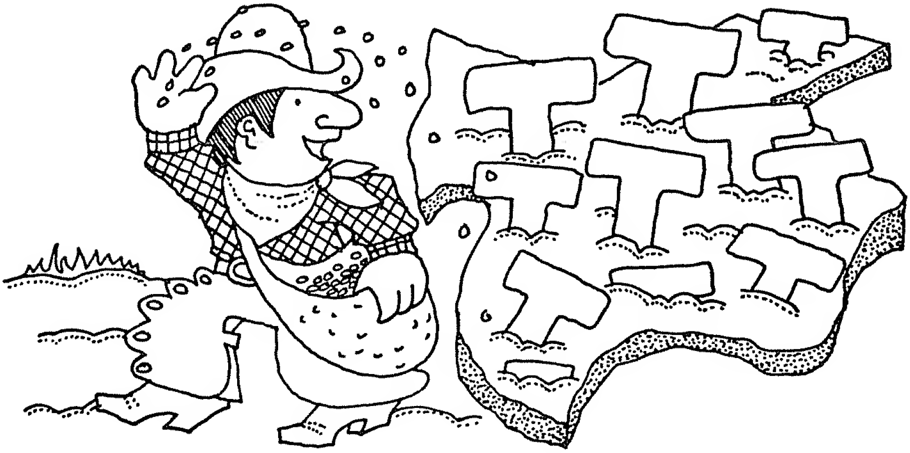
You might make it possible for the program to continue when a duplicate file name is used or when a file name cannot be found.

KNOWLEDGE



31

AL'S TOUR OF THE STATES



For Parents and Teachers ...

This game teaches children the names of all the states. A letter appears on the screen, and the children have to guess which state begins with that letter. If they can't think of any states or they can't spell the state name, the computer will let them peek at the state names beginning with each letter.

For Kids ...

Here's Alphabet Al sowing "alphabet" seeds all over the country. When Al visits a state he finds out the first letter in the state's name and then plants only that letter. For example, Al plants T's all over Texas, A's all over Alaska, and N's all over New Jersey.

Al doesn't plant any B's, E's, J's, Q's, X's, Y's, or Z's, since there are no states beginning with those letters.

After Al enters a new state and plants a letter, he stops and asks you to guess what state he is in. If you guess a state that begins with the letter Al has just planted, you win. If not, Al lets you try again.

If you can't think of a state beginning with the right letter, just press the **ENTER** button. Al will flash all the states beginning with that letter on the TV screen. When you're through studying the state names, just press **ENTER** again (or any other key). Then Al gives you another chance.

The Game ...

Program Name: **STATES**

```
10 REM *** AL'S TOUR OF THE STATES
12 CLEAR 500
50 CLS
65 PRINT"*** AL'S TOUR OF THE STATES ***
"
66 FOR PAUSE=1 TO 1000: NEXT PAUSE
70 FOR X=65 TO 90
75 IF X=66 OR X=69 OR X=74 OR X=81 OR X>
87 THEN 168
120 CLS
125 FOR I=1 TO 25
130 P=RND(350):PRINT@P,CHR$(X);
140 SOUND RND(87)+88,1
142 NEXT I
145 FOR PAUSE=1 TO 250:NEXT PAUSE
146 CLS
150 PRINT@288,"WHAT STATE IS AL IN";:INP
UT A$
154 IFA$="" THEN CLS:GOSUB 3010:GOTO 120
156 RESTORE
157 FOR STATE=1 TO 50:READ B$:IF B$<>A$
THEN NEXT STATE
158 IF B$=A$ AND ASC(LEFT$(A$,1))=X THEN
 GOSUB 1010:GOTO 168
164 GOSUB 2010:GOTO 120
```

```
168 NEXT X
170 CLS:END
3000 REM *** DISPLAY STATES
3010 PRINT@32,"STATES BEGINNING WITH THE
 LETTER ";CHR$(X)
3020 PRINT"============================
=="
3022 PRINT
3025 RESTORE:FOR NUM=1 TO 50:READ B$:IF
ASC(LEFT$(B$,1))=X THEN PRINT"    ";B$
3028 NEXT NUM
3100 A$=INKEY$:IF A$="" THEN 3100
3140 RETURN
4000 DATA ALABAMA,ALASKA,ARIZONA,ARKANSA
S,CALIFORNIA
4005 DATA COLORADO,CONNECTICUT,DELAWARE,
FLORIDA,GEORGIA
4010 DATA HAWAII,IDAHO,ILLINOIS,INDIANA,
IOWA
4015 DATA KANSAS,KENTUCKY,LOUISIANA,MAIN
E,MARYLAND
4020 DATA MASSACHUSETTS,MICHIGAN,MINNESO
TA,MISSISSIPPI,MISSOURI
4025 DATA MONTANA,NEBRASKA,NEVADA,NEW HA
MPSHIRE,NEW JERSEY
4030 DATA NEW MEXICO,NEW YORK,NORTH CARO
LINA,NORTH DAKOTA,OHIO
4035 DATA OKLAHOMA,OREGON,PENNSYLVANIA,R
HODE ISLAND,SOUTH CAROLINA
4040 DATA SOUTH DAKOTA,TENNESSE,TEXAS,UT
AH,VERMONT
4045 DATA VIRGINIA,WASHINGTON,WEST VIRGI
NIA,WISCONSIN,WYOMING
```

Highlights ...

On line 75, the program checks for letters that do not begin state names: B's, E's, J's, Q's, X's, Y's, or Z's. If X equals the ASCII code for that number, the program jumps to line 168 and a new code (the next letter in the alphabet) is chosen.

On line 150, the program asks what state Al is in. If you just press **ENTER**, then the program jumps to the Display States subroutine on line 3010.

On lines 3010 and 3020, the program prints out a title. Next (on lines 3025 and 3028), the program searches through the state names listed in the DATA statements on lines 4000 to 4070. The names are in alphabetical order. When the program arrives at the first state name beginning with Alphabet Al's current letter, it prints that name.

On line 3028, the program continues searching through the list of state names. If it finds more names that begin with Al's letter, it prints those out, too.

The list of names stays on the screen until you press **ENTER** (or any other button). The program takes care of this on line 3100. On line 3100, it looks at A$. If you haven't pressed a button yet, A$ stays blank. After you press any button, A$ will change. Line 3100 is a tiny loop. The computer keeps checking until it changes (you press a button). Line 3140 turns the computer's attention away from the keyboard and back to the program.

When you enter the name of a state, it gets stored in A$. The computer checks its list of state names (DATA statements 4000 to 4045) on lines 157 and 158.

If there is a match between your answer (A$) and the state name the computer pulls from the list (B$), then (on line 158) the computer jumps to the Happy Face subroutine on line 1010.

If the computer can't find a match (on lines 157 or 158), it jumps to the Sad Face subroutine. Remember to LOAD the Happy and Sad Face subroutine (FACES) before typing this program.

Variables ...

A$	Your answer—what state Al is in.
B$	A state from the computer's list.
X	Counter—the ASCII code for the letters in the alphabet.

I — Counter—Al plants 25 letters in each state.

STATE — Counter—computer searches through list of 50 states to find a match with your answer.

NUM — Counter—computer searches through list of 50 states to display states beginning with certain letter.

Do-It-Yourself ...

Al runs through the alphabet from A to Z. You can make him run through the alphabet backwards, or you can have him hop around the alphabet.

Hint #1: Look at the FOR-NEXT loop on line 70. Hint #2: A FOR-NEXT loop can go backwards. In the FOR command, remember to include STEP −1. This makes the counter go backwards from the higher number to the lower number, one number at a time.

How would you make Al print letters at random?

Hint #1: You need to change lines 70 and 168. Hint #2: To compute a number representing a random letter of the alphabet, use the command RND(26)+64. This gives you a random number between 65 and 90 (the ASCII codes representing the upper-case letters).

If Al's pace is too frantic and you'd like to slow him down a bit, you can add a delay loop at line 141 by typing between lines 140 and 142 this line:

```
141 FOR D=1 TO 200:NEXT D
```

You can also increase the amount of time the letters are on the screen before Al erases them and pops his question. Just increase the PAUSE loop on line 145 from 250 to 500 (or even higher).

Here is another idea. Why not have Al plant things other than letters? Letters would still fly around the screen, but they would stand for foreign countries, fruits, sports cars, rock stars, or whatever.

Hint: You will need to change the title on lines 10 and 65. You will need to change the question on line 150. You will need to change the titles on lines 3000 and 3010. You will need to replace the DATA statements on lines 4000 to 4045. You will also need to change the FOR statements on lines 157 and 3025 to the number of pizza toppings, animals, or whatever you are having Al plant (the old number is 50 for the 50 states listed in the DATA statements).

32

WHAT'S YOUR NUMBER?

For Parents and Teachers ...

This game will help younger children learn their telephone number.

For Kids ...

The computer is going to help you learn your phone number.

When you type RUN, the phone game begins. The computer asks you for your name and your phone number. Your mom or dad can write it down on a piece of paper. Then you can type the number on the computer's typewriter. Then hide the paper—and don't peek.

The computer will ask you for the number again. Can you remember it?

If you can't remember, try guessing. After three wrong guesses, the computer will let you peek at the number. Then it will hide it and ask if you want to play the game again. Type "YES" and see if you can get the number right.

If it takes you a long time, don't worry. The computer never loses its patience and never gets angry. It won't growl at you or stick out its tongue. It keeps playing the game until you know your phone number perfectly.

The Game . . .

Program Name: **PHONE**

```
5 REM *** LEARN YOUR PHONE NUMBER
10 REM THIS PROGRAM HELPS YOUNGER CHILDR
EN TO LEARN THEIR PHONE NUMBER
20 REM MELISSA PERDUE
50 CLS
55 PRINT@34,"***  PHONE NUMBER GAME  ***
"
56 FOR PAUSE=1 TO 1000:NEXT PAUSE
60 REM N$ STORES PLAYER'S NAME
70 REM SPH$ STORES PHONE NUMBER
80 REM FPH$ STORES PHONE NUMBER AS
82 REM REMEMBERED WITHOUT AID
120 CLS
130 PRINT@77,"HELLO!"
145 PRINT@225,"WHAT IS YOUR NAME";:INPUT
 N$
160 CLS
165 PRINT@226,"HI, THERE, ";N$;"!"
170 PRINT@290,"WHAT IS YOUR"
185 PRINT@321," ";
186 PRINT@322,"PHONE NUMBER";:INPUT SPH$
250 CLS
260 PRINT@226,"CAN YOU REMEMBER"
270 PRINT@290,"ALL THAT?"
275 PRINT@354,"LET'S TRY!!!"
280 FOR DELAY=1 TO 2000:NEXT DELAY
282 COUNT=0
285 CLS
287 PRINT@226,"WHAT IS YOUR"
```

```
288 PRINT@290,"PHONE NUMBER";:INPUT FPH$
300 IF SPH$=FPH$ THEN CLS:GOSUB 1010:GOT
O 350
310 COUNT=COUNT+1
320 IF COUNT>=3 THEN 340
330 CLS:GOSUB 2010:GOTO 285
340 CLS:PRINT@225," ";
345 PRINT"YOUR PHONE NUMBER IS:"
346 PRINT@290,SPH$
347 FOR DELAY=1 TO 3000:NEXT DELAY
350 CLS
355 PRINT@226,"PLAY AGAIN";:INPUT A$
360 IF LEFT$(A$,1)="Y" THEN 282
370 IF LEFT$(A$,1)<>"N" THEN 350
380 CLS:END
```

Highlights . . .

The What's Your Number? phone game was developed by Melissa Perdue, a student at Patrick Henry High School in Roanoke, Virginia.

The phone game program stores the phone number in the variable SPH$. When you enter your guess, it stores that in FPH$. If FPH$ matches SPH$ then the happy face appears. If not, the wrong-answer counter, count, is incremented by 1, and the sad face appears. If COUNT is greater than or equal to 3, then the computer prints the phone number for you to study. Then it asks if you would like to play the phone game again. Be sure to LOAD in the FACES subroutine before typing in this program.

Variables . . .

PAUSE	Delay loop counter.
N$	Your name.
SPH$	Phone number.

FPH$	Phone number guess.
A$	Answer to the question “PLAY AGAIN?”

Do-It-Yourself ...

Your children can use this game to learn their seven-digit phone number or their 10-digit phone number including the area code.

With just a few changes, they can also use this game to learn other phone numbers—their friends' numbers, police, fire department, grandparents, movie theaters, etc.

Right now, if the children get their number wrong three times, the program displays the correct number. You might want to change the program to have the correct number displayed after one wrong answer.

33

WHERE DO YOU LIVE?

For Parents and Teachers . . .

This game helps children learn their address, including their street, apartment, city, state, and zip code. It teaches them how to format their address on an envelope.

For Kids . . .

This game helps you learn your address and how to write your address on an envelope when you send a letter.

The computer asks you for your first name. Next it asks for your last name. Then it prints out your name and address, just the way you would write them on an envelope to mail to your grandparents or to a friend.

Then the computer asks you to type in your address, one part at a time. It asks you for your street, your apartment (if you live in an apartment), your city, your state, and your zip code.

If you can't remember part of your address, don't worry. Try guessing. The computer will give you three chances, then it will let you peek at the part of the address you can't remember. Then it will hide it and ask you to type it in yourself.

Maybe some parts will take dozens of tries to get just right. But the computer won't lose its temper or grumble. It never gets upset. It just keeps playing the game until you know your whole address.

The Game . . .

Program Name: **ADDRESS**

```
 5 REM *** LEARN YOUR ADDRESS
 10 REM THIS PROGRAM HELPS YOUNGER CHILDR
 EN LEARN THEIR ADDRESS
 20 REM MELISSA PERDUE
 50 CLS
 55 PRINT@35,"*** LEARN YOUR ADDRESS ***"
 56 FOR PAUSE=1 TO 2000:NEXT PAUSE
 60 CLEAR 500
 70 FOR J=1 TO 5:READ AD$(J):NEXT J
 80 FOR J=1 TO 5:READ RA$(J):NEXT J
 91 RESTORE
 130 PRINT@109,"HELLO!"
 150 PRINT@386,"WHAT IS YOUR:"
➧155 PRINT@421,"first name";:INPUT N1$
 158 PRINT@421," "
➧159 PRINT@421,"last name";:INPUT N2$
 160 N$=N1$+" "+N2$
 164 CLS
 167 PRINT@64,"HI, THERE, ";N$;"!"
 168 GO SUB 3010:REM * DISPLAY ADDRESS
 170 CLS
 172 FOR J=1 TO 5
 180 IF RA$(J)="" THEN 240
 187 CLS
 188 PRINT@194,"WHAT IS YOUR"
 194 PRINT@207,AD$(J):INPUT GAD$
 200 IF GAD$=RA$(J) THEN GOSUB 1010:GOTO
 240
 210 COUNT=COUNT+1:GOSUB 2010:IF COUNT <3
 THEN 187
215 CLS
220 PRINT@194,"YOUR ";AD$(J);" IS"
225 PRINT@226, RA$(J)
230 FOR PAUSE=1 TO 1500:NEXT PAUSE
232 COUNT=0
235 GOTO 187
240 NEXT J
250 GOSUB 3010:REM * DISPLAY ADDRESS
260 CLS
270 PRINT@386,"PLAY AGAIN (Y/N)";:INPUT
A$
280 IF A$="Y" THEN 170
290 IF A$<>"N" THEN 270
300 CLS:END
3000 REM *** DISPLAY ADDRESS
3010 CLS
3020 PRINT@74,"YOUR ADDRESS"
3021 PRINT@106,"-----------"
3025 PRINT@164,N$
3030 PRINT@196,RA$(1);"  ";
3035 PRINTRA$(2)
3036 PRINT@228,RA$(3);", ";
3040 PRINTRA$(4);"  ";
3050 PRINTRA$(5)
3080 FOR PAUSE=1 TO 3000:NEXT PAUSE
3090 RETURN
5001 DATA STREET OR ROAD
5002 DATA APARTMENT
5003 DATA CITY
5004 DATA STATE
5005 DATA ZIP
6001 DATA 2117 CARTER ROAD SW
6002 DATA
6003 DATA ROANOKE
6004 DATA VIRGINIA
6005 DATA 24015
```

Typing Hints . . .

Remember to LOAD the Happy and Sad Face subroutine (FACES) before typing this program.

Lines 155 and 159 contain lower-case letters. This means they are to be typed in reverse video. To make the first and last name message display in reverse video, you need to press the **SHIFT** and **0** keys simultaneously. To turn it off, just press the **SHIFT** and **0** keys again.

Highlights ...

The Where Do You Live address game was developed by Melissa Perdue, a student at Patrick Henry High School in Roanoke, Virginia.

The address game program has a sample (Roanoke) address entered on lines 6001 to 6005. You will need to replace this with your address. Your street goes on line 6001. Your apartment goes on 6002. If you don't live in an apartment, just leave the empty DATA command at 6002. Your city goes on line 6003, your state at 6004, your zip code at 6005.

The address categories are stored in DATA statements on lines 5001 to 5005 (i.e., STREET OR ROAD, APARTMENT, CITY, STATE, and ZIP). The program READs in the address categories and the actual addresses into string arrays on lines 70 and 80. If a category is empty (for example, your apartment address), the program skips to the next category.

The main action in the program takes place inside the loop from lines 172 to 240. Each time the child gets a wrong answer, the wrong-answer counter, COUNT, is incremented by 1. If COUNT is less than 3, the computer keeps asking for the same part of his or her address. When COUNT reaches 3, the computer displays the correct answer, then erases it and lets the child try again.

At the beginning and end of the game, the program calls the subroutine beginning on line 3000. This subroutine displays the child's name and address in the format in which it would appear on an envelope.

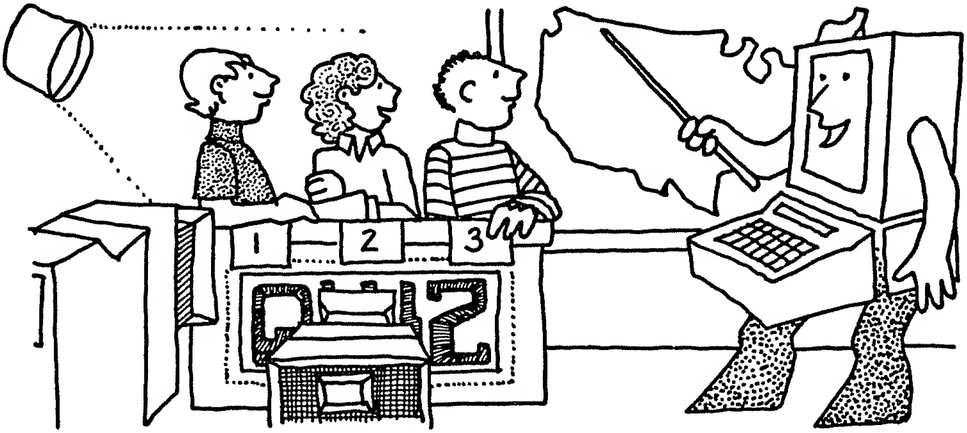
Variables ...

PAUSE	Delay loop counter.
N$	Your name.
N1$	First name.
N2$	Last name.
A$	Answer to question "PLAY AGAIN?"
GAD$	Address guess.
AD$	Array—address category (street, city, etc.).
RA$	Array—address (your street name, city name, etc.).
J	Loop counter—main loop of game.
COUNT	Wrong-answer counter.

Do-It-Yourself ...

With just a couple of changes, this game can help children learn other people's addresses—friends, grandparents, etc.

34
QUIZ SHOW



For Parents and Teachers ...

This game helps children learn some interesting facts. The questions in this game can be easily replaced by questions focusing on a particular subject: history, geography, current events, language arts, etc. Also, many new questions can be added.

The game is self-grading. It keeps track of the children's correct and incorrect answers. The children's score is printed at the end of the quiz.

For Kids ...

Pretend that you get a letter in the mail. You have been invited to be on a TV quiz show!

You go to the library, and take out dozens of books. You read the newspaper every day. You study your encyclopedia. You feel so full of facts and figures that they might soon pop out of your ears!

The big day arrives. You go to the TV studio. The show begins. They put makeup on your face, so you don't look pale under the cameras. The stage is hot. The lights are bright. You feel nervous. Here comes the first question: Who is the father of our country? George Washington? John Adams? Thomas Jefferson?

For a moment, you throw a blank. None of the names sounds right. Then you recover. "George Washington?" you say.

The quiz show announcer's happy face appears. "Right!" he says. Happy music floods the studio. The other kids look at you enviously. How could you be so sharp, so cool under pressure?

You relax. This is going to be easy.

The Game . . .

Program Name: **QUIZ**

```
50 REM *** QUIZ SHOW
55 REM DESIGNED BY SCOTT RAINEY AND BRIA
N FRANCOIS
60 CLS
65 PRINT@37,"***   QUIZ SHOW   ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
120 CLS:PRINT@34,"WHO IS THE FATHER OF O
UR      COUNTRY?":PRINT
140 PRINT" A) GEORGE WASHINGTON"
142 PRINT" B) JOHN ADAMS"
144 PRINT" C) THOMAS JEFFERSON"
150 PRINT:INPUT A$
151 IF A$="A" THEN RC=RC+1:GO SUB 1010:G
OTO 155
152 WC=WC+1:GOSUB 2010
155 CLS:PRINT@34,"WHAT IS THE CAPITAL OF
 THE    UNITED STATES?":PRINT
170 PRINT" A) NEW YORK"
```

```
172 PRINT" B) WASHINGTON D.C."
174 PRINT" C) PHILADELPHIA"
180 PRINT:INPUT A$
181 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 185
182 WC=WC+1:GOSUB 2010
185 CLS:PRINT@34,"WHO IS THE CURRENT PRE
SIDENT  OF THE UNITED STATES?":PRINT
200 PRINT" A) JIMMY CARTER"
202 PRINT" B) RICHARD NIXON"
204 PRINT" C) RONALD REAGAN"
210 PRINT:INPUT A$
211 IF A$="C" THEN RC=RC+1:GOSUB 1010:GO
TO 215
212 WC=WC+1:GOSUB 2010
215 CLS:PRINT@34,"WHICH COUNTRY GIVES IT
S PEOPLE THE MOST FREEDOM?":PRINT
230 PRINT" A) RUSSIA"
232 PRINT" B) POLAND"
234 PRINT" C) UNITED STATES"
240 PRINT:INPUT A$
241 IF A$="C" THEN RC=RC+1:GOSUB 1010:GO
TO 245
242 WC=WC+1:GOSUB 2010
245 CLS:PRINT@34,"WHICH COUNTRY WAS THE
FIRST TO PUT A MAN IN SPACE?":PRINT
260 PRINT" A) RUSSIA"
262 PRINT" B) UNITED STATES"
264 PRINT" C) CHINA"
270 PRINT:INPUT A$
271 IF A$="A" THEN RC=RC+1:GOSUB 1010:GO
TO 275
272 WC=WC+1:GOSUB 2010
275 CLS:PRINT@34,"WHO DISCOVERED ELECTRI
CITY?":PRINT
290 PRINT" A) BENJAMIN FRANKLIN"
292 PRINT" B) ISAAC NEWTON"
294 PRINT" C) THOMAS JEFFERSON"
300 PRINT:INPUT A$
301 IFA$="A" THEN RC=RC+1:GOSUB 1010:GOT
O 305
302 WC=WC+1:GOSUB 2010
305 CLS:PRINT@34,"WHO DISCOVERED AMERICA
 IN 1492?":PRINT
320 PRINT" A) AMERIGO VESPUCCI"
```

```
322 PRINT" B) VIKINGS"
324 PRINT" C) CHRISTOPHER COLUMBUS"
330 PRINT:INPUT A$
331 IF A$="C" THEN RC=RC+1:GOSUB 1010:GO
TO 335
332 WC=WC+1:GOSUB 2010
335 CLS:PRINT@34,"WHAT COUNTRY WAS THE F
IRST TO PUT A MAN IN THE MOON?":PRINT
350 PRINT" A) RUSSIA"
352 PRINT" B) UNITED STATES"
354 PRINT" C) BRITAIN"
360 PRINT:INPUT A$
361 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 365
362 WC=WC+1:GOSUB 2010
365 CLS:PRINT@34,"WHAT WAS THE FIRST COU
NTRY TO ENTER THE NUCLEAR AGE?":PRINT
370 PRINT" A) UNITED STATES"
372 PRINT" B) CHINA"
374 PRINT" C) RUSSIA"
390 PRINT:INPUT A$
391 IF A$="A" THEN RC=RC+1:GOSUB 1010:GO
TO 395
392 WC=WC+1:GOSUB 2010
395 CLS:PRINT@34,"WHAT WAR WON THE UNITE
D STATES HER FREEDOM?":PRINT
400 PRINT" A) WAR OF 1812"
402 PRINT" B) REVOLUTIONARY WAR"
404 PRINT" C) CIVIL WAR"
420 PRINT:INPUT A$
421 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 425
422 WC=WC+1:GOSUB 2010
425 CLS:PRINT@34,"WHO INVENTED THE TELEP
HONE?":PRINT
440 PRINT" A) THOMAS EDISON"
442 PRINT" B) ALEXANDER GRAHAM BELL"
444 PRINT" C) BENJAMIN FRANKLIN"
450 PRINT:INPUT A$
451 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 455
452 WC=WC+1:GOSUB 2010
455 CLS:PRINT@34,"WHEN IS THE AMERICAN
        INDEPENDENCE DAY?":PRINT
470 PRINT" A) NOVEMBER 25"
```

```
472 PRINT" B) OCTOBER 31"
474 PRINT" C) JULY 4"
480 PRINT:INPUT A$
481 IF A$="C" THEN RC=RC+1:GOSUB 1010:GO
TO 485
482 WC=WC+1:GOSUB 2010
485 CLS:PRINT@34,"WHO WAS PRESIDENT OF T
HE      UNITED STATES AT THE BEGINNING
OF THE CIVIL WAR?":PRINT
500 PRINT" A) ABRAHAM LINCOLN"
502 PRINT" B) ANDREW JACKSON"
504 PRINT" C) U. S. GRANT"
510 PRINT:INPUT A$
511 IF A$="A" THEN RC=RC+1:GOSUB 1010:GO
TO 515
512 WC=WC+1:GOSUB 2010
515 CLS:PRINT@34,"WHICH OF THE FOLLOWING
 IS A   NUMERIC VARIABLE?":PRINT
530 PRINT" A) AX$"
532 PRINT" B) Z"
534 PRINT" C) 45"
540 PRINT:INPUT A$
541 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 545
542 WC=WC+1:GOSUB 2010
545 CLS:PRINT@34,"WHAT BRAND OF COMPUTER
 ARE YOU USING?":PRINT
550 PRINT" A) ATARI"
551 PRINT" B) RADIO SHACK"
552 PRINT" C) TIMEX"
570 PRINT:INPUT A$
571 IF A$="B" THEN RC=RC+1:GOSUB 1010:GO
TO 575
572 WC=WC+1:GOSUB 2010
575 CLS:PRINT@226,"YOUR SCORE IS:"
590 PRINT:PRINT
600 PRINT"   ";RC;"RIGHT"
610 PRINT"   ";WC;"WRONG"
620 PRINT:PRINT
630 IF RC=15 THEN PRINT"YOU ARE A GENIUS
!":GOTO 700
640 IF RC>=11 THEN PRINT"YOU ARE PRETTY
SMART!":GOTO 700
650 IF RC=>8 THEN PRINT"YOU DID OKAY.":G
OTO 700
```

```
660 IF RC=>4 THEN PRINT"YOU NEED A LITTL
E WORK.":GOTO 700
670 IF RC=>1 THEN PRINT"YOU NEED TO STUD
Y MORE!":GOTO 700
680 PRINT"OH! OH! OH!"
700 FOR PAUSE=1 TO 3000:NEXT PAUSE
710 CLS:END
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (FACES) before typing in this program. Change line 2150 to look like this:

```
2150 PRINT@486,"SORRY ... THAT'S WRONG";
```

Background ...

The Quiz Show game was designed by Scott Rainey and Brian Francois, students at Patrick Henry High School in Roanoke, Virginia.

The boys had only two weeks to put this program together. Also, they had to use a faulty program recorder. At the end of each day they would save the current version of their program. The next morning they would try to load the program back into the computer. But the program had disappeared. The recorder had "eaten" it!

In order to complete the program on time, the boys had to get permission to skip some of their classes at school. They finally completed the program on the day it was due.

They write: "In order to complete this program, we had to overcome many obstacles. We hope that your child will learn and benefit from our work."

Highlights ...

The CLS at the beginning of each question causes the computer to clear the screen.

There are two counters. The RC counter keeps track of correct answers. The WC counter keeps track of the wrong answers. At the end of the quiz, your score is printed along with a rating. The rating goes from "YOU'RE A GENIUS!" (for all 15 questions correct) to "OH! OH! OH!" (for all 15 questions wrong).

Line 2150 in the Sad Face subroutine is changed since the program does not permit retries after an incorrect answer.

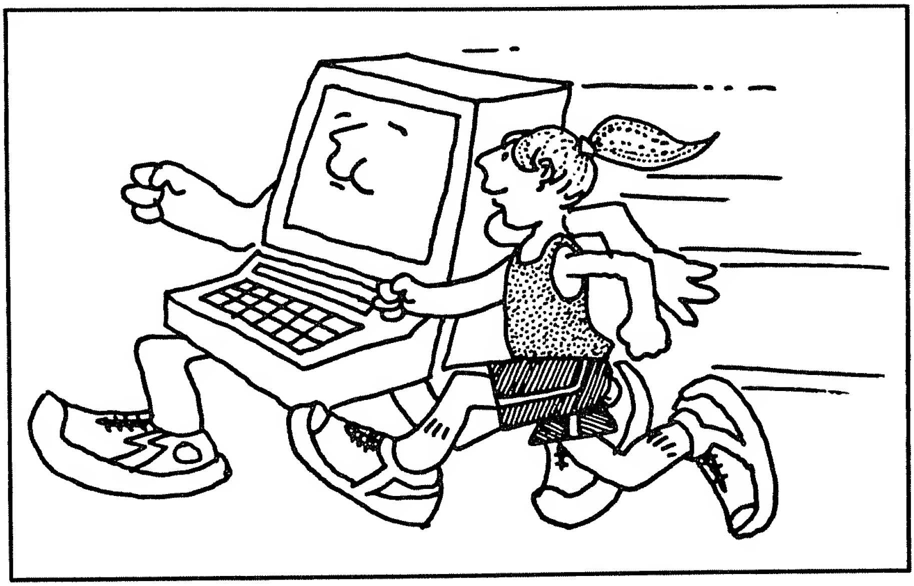
Variables . . .

PAUSE	Delay loop counter.
A$	Accepts your one-letter answer to quiz show questions.
RC	Counter for right answers.
WC	Counter for wrong answers.

Do-It-Yourself . . .

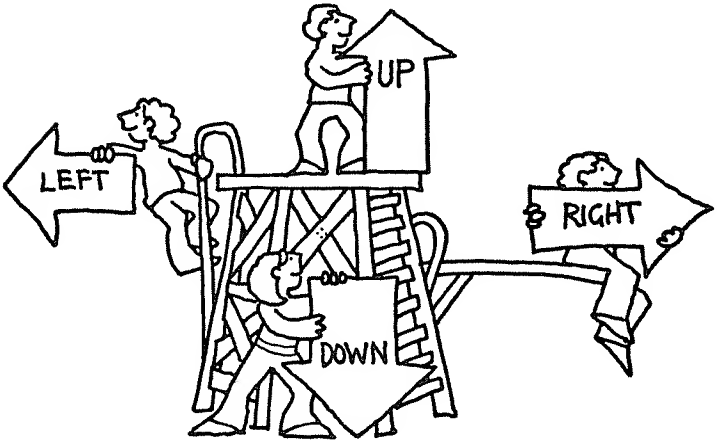
These questions are just a sample of the types of questions you can think up for the quiz show game. You can set up a quiz show based on your favorite facts or trivia. Do you like movies? Sports? Astronomy? Adventure games? Mysteries? Rock music? You can set up a quiz show on any of these subjects, or on anything you want.

HAND-EYE



35

UP-DOWN-LEFT-RIGHT



For Parents and Teachers ...

This game helps children learn directions: up, down, left, and right. It also helps them recognize the direction words.

For Kids ...

Look for the arrow keys on the keyboard. Do you see them? Now, you're ready.

The computer asks you: "WHICH DIRECTION?"

On the screen you see an arrow and a word. What arrow do you see? It's the up arrow. The arrow is pointing to the word "UP." Say the word out loud. Shout it out!

Can you find the same arrow on the computer keyboard? Press it. Hurray!

Now a different arrow and a different word appear. This is the down arrow and the word "DOWN." Shout out the word

"DOWN!" Find the down arrow on the keyboard and press it.

Next, a different arrow and a different word appear. This is the left arrow and the word "LEFT." Shout out the word "LEFT!" Find the left arrow on the keyboard and press it.

One last arrow and one last word appear. This is the right arrow and the word "RIGHT." Shout out the word "RIGHT!" Find the right arrow on the keyboard and press it.

The game is over. The computer will ask you if you want to play again. You should type in "YES" or "Y." This time you are on your own. Good luck!

The Game . . .

Program Name: **UPDOWN**

```
50 REM *** UP/DOWN GAME
60 CLS
65 PRINT@34,"***  UP-DOWN-LEFT-RIGHT  **
*"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 DIM B(4):B(1)=94:B(2)=10:B(3)=8:B(4)=
9
110 FOR D=1 TO 4
111 CLS
120 PRINT@4,"*** WHICH DIRECTION? ***"
130 ON D GOSUB 3010,3510,4010,4510
150 K$=INKEY$:IF K$="" THEN 150
160 IF ASC(K$)=B(D) THEN CLS:GOSUB 1010:
GOTO 200
170 CLS:GOSUB 2010:GOTO 111
200 NEXT D
210 CLS
220 PRINT@226,"AGAIN";:INPUT A$
230 IF LEFT$(A$,1)="Y" THEN 110
240 IF LEFT$(A$,1)<>"N" THEN 210
250 CLS:END
3000 REM *** UP SUBROUTINE
➧3010 PRINT@46,"u p"
3020 PRINT@175,CHR$(128);
3040 PRINT@141,CHR$(139);
3050 PRINT@145,CHR$(135);
```

```
  3060 PRINT@110,CHR$(137);
  3070 PRINT@112,CHR$(134);
  3080 PRINT@79,CHR$(140);
  3090 PRINT@143,CHR$(128);
  3100 RETURN
  3500 REM *** DOWN SUBROUTINE
➧ 3510 PRINT@45,"down"
  3520 PRINT@79,CHR$(140);
  3530 PRINT@111,CHR$(128);
  3540 PRINT@109,CHR$(142);
  3550 PRINT@113,CHR$(141);
  3560 PRINT@142,CHR$(134);
  3570 PRINT@144,CHR$(137);
  3580 PRINT@175,CHR$(131);
  3590 RETURN
  4000 REM *** LEFT SUBROUTINE
➧ 4010 PRINT@45,"left"
  4020 PRINT@79,CHR$(141);
  4030 PRINT@110,CHR$(137);
  4040 PRINT@141,CHR$(138);
  4050 PRINT@143,CHR$(128);
  4060 PRINTCHR$(128);CHR$(128);
  4070 PRINT@174,CHR$(134);
  4080 PRINT@207,CHR$(135);
  4090 RETURN
  4500 REM *** RIGHT SUBROUTINE
➧ 4510 PRINT@45,"right"
  4520 PRINT@141,CHR$(128);
  4530 PRINTCHR$(128);CHR$(128);
  4540 PRINT@145,CHR$(133);
  4550 PRINT@176,CHR$(137);
  4560 PRINT@207,CHR$(139);
  4570 PRINT@79,CHR$(142);
  4580 PRINT@112,CHR$(134);
  4590 RETURN
```

Typing Hints . . .

You get two different colors on the screen by entering the direction names in reverse video. These appear in the listing as lower-case letters. To turn on the reverse video, just press the **SHIFT** and **0** keys. To turn it off, press the **SHIFT** and **0** keys again.

Highlights ...

All four directions run inside of the loop on lines 110 to 200. The "up" screen is printed by the subroutine beginning on line 3000. The "down" screen is printed by the subroutine beginning on line 3500. The "left" screen is printed by the subroutine beginning on line 4000. The "right" screen is printed by the subroutine beginning on line 4500.

Now look at lines 3020, 3520, 4020, and 4520. These lines print the different arrows. You get the arrows to appear on the screen by using the PRINT CHR$(N) command, where N represents the graphic character to be printed.

The arrow keys are represented by the B array. Up arrow = ASCII code 94, down arrow = ASCII code 10, left arrow = ASCII code 8, right arrow = ASCII code 9.

Variables ...

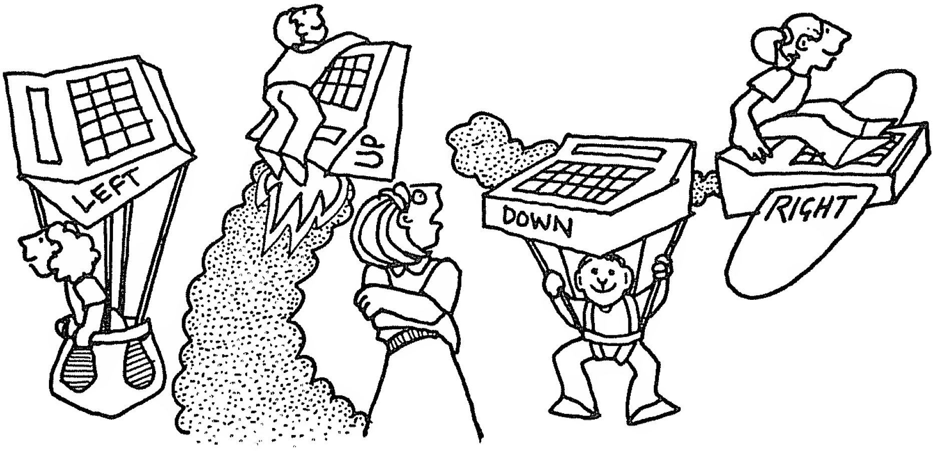
PAUSE	Delay loop counter.
A$	Accepts your answer to question "PLAY AGAIN?"
B	Array—contains values for arrow keys.
D	Direction loop counter (for 4 directions).
K$	Accepts arrow value.

Do-It-Yourself ...

There are lots of ways to spice up this program. But before you start, turn to the next chapter.

36

UP, UP AND AWAY!



For Parents and Teachers ...

Like the game in the last chapter, this game helps children learn directions—up, down, left, and right—and direction names.

This game is an expanded version of the up-down game from the last chapter. It shows what happens when you add animation and sound effects to the game.

For Kids ...

This game is a lot like the game in the last chapter. Only this time the arrows move and play funny music.

To play this game, just search for the arrows on the computer keyboard. Press the arrow that matches the arrow on the TV screen. Remember to shout out the directions: "UP!" "DOWN!" "LEFT!" and "RIGHT!"

The Game ...

Program Name: **UPDOWN2**

```
50 REM *** UP/DOWN GAME
60 CLS
65 PRINT@34,"***  UP-DOWN-LEFT-RIGHT  **
*"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 CLEAR 400
80 DIM H(4,4)
90 DIM B(4):B(1)=94:B(2)=10:B(3)=8:B(4)=
9
110 FOR D=1 TO 4
111 CLS
120 PRINT@4,"*** WHICH DIRECTION? ***"
130 ON D GOSUB 3010,3510,4010,4510
150 K$=INKEY$:IF K$="" THEN 150
160 IF ASC(K$)=B(D) THEN CLS:GOSUB 1010:
GOTO 200
170 CLS:GOSUB 2010:GOTO 111
200 NEXT D
210 CLS
220 PRINT@226,"AGAIN";:INPUT A$
230 IF LEFT$(A$,1)="Y" THEN 110
240 IF LEFT$(A$,1)<>"N" THEN 210
250 CLS:END
3000 REM *** UP SUBROUTINE
3010 PRINT@46,"u p"
3020 PRINT@175,CHR$(128);
3040 PRINT@141,CHR$(139);
3050 PRINT@145,CHR$(135);
3060 PRINT@110,CHR$(137);
3070 PRINT@112,CHR$(134);
3080 PRINT@79,CHR$(140);
3090 PRINT@143,CHR$(128);
3100 FOR DELAY=1 TO 500:NEXT
3112 P=1389:COUNT=0
3115 GOSUB 5000
3120 FOR I=4 TO 0 STEP-1
3130 FOR J=0 TO 4
3140 POKE I*32+J+P,H(I,J)
3150 NEXT J
3155 SOUND 192-(I*8),1:NEXT I
```

```
3160 FOR DELAY=1 TO 200:NEXT
3161 COUNT=COUNT+1:IF COUNT=> 9 THEN RET
URN
3170 P=P-32
3190 GOTO 3120
3500 REM *** DOWN SUBROUTINE
➧3510 PRINT@45,"down"
3520 PRINT@79,CHR$(140);
3530 PRINT@111,CHR$(128);
3540 PRINT@109,CHR$(142);
3550 PRINT@113,CHR$(141);
3560 PRINT@142,CHR$(134);
3570 PRINT@144,CHR$(137);
3580 PRINT@175,CHR$(131);
3590 GO SUB 5010
3612 P=1133:COUNT=0
3620 FOR I=0 TO 4
3630 FOR J=0 TO 4
3640 POKE I*32+J+P,H(I,J)
3650 NEXT J
3655 SOUND I*8+170,1:NEXT I
3660 FOR DELAY=1 TO 200:NEXT
3661 COUNT=COUNT+1:IF COUNT=>9 THEN RETU
RN
3665 POKE P+2,96
3670 P=P+32
3690 GOTO 3620
4000 REM *** LEFT SUBROUTINE
➧4010 PRINT@45,"left"
4020 PRINT@79,CHR$(141);
4030 PRINT@110,CHR$(137);
4040 PRINT@141,CHR$(138);
4050 PRINT@143,CHR$(128);
4060 PRINTCHR$(128);CHR$(128);
4070 PRINT@174,CHR$(134);
4080 PRINT@207,CHR$(135);
4090 FOR DELAY=1 TO 200:NEXT
4112 P=1100:COUNT=0
4115 GOSUB 5000
4120 FOR I=0 TO 4
4130 FOR J=O TO 4
4140 POKE I*32+J+P,H(I,J)
4150 NEXT J
4155 SOUND I*8+170,1:NEXT I
4160 FOR DELAY=1 TO 200:NEXT
4161 COUNT=COUNT+1:IF COUNT=> 9 THEN RET
```

```
URN
4165 POKE P+68,96
4170 P=P-1
4190 GOTO 4120
4500 REM *** RIGHT SUBROUTINE
4510 PRINT@45,"right"
4520 PRINT@141,CHR$(128);
4530 PRINTCHR$(128);CHR$(128);
4540 PRINT@145,CHR$(133);
4550 PRINT@176,CHR$(137);
4560 PRINT@207,CHR$(139);
4570 PRINT@79,CHR$(142);
4580 PRINT@112,CHR$(134);
4590 FOR DELAY=1 TO 200:NEXT
4612 P=1102:COUNT=0
4615 GOSUB 5000
4620 FOR I=4 TO 0 STEP-1
4630 FOR J=0 TO 4
4640 POKE I*32+J+P,H(I,J)
4650 NEXT J
4655 SOUND 192-(I*8),1:NEXT I
4660 FOR DELAY=1 TO 200:NEXT
4661 COUNT=COUNT+1:IF COUNT=>9 THEN RETU
RN
4665 POKE P+64,96
4670 P=P+1
4690 GOTO 4620
5000 REM CAPTURE ARROW
5010 FOR I=0 TO 4
5020 FOR J=0 TO 4
5030 H(I,J)=PEEK(32*I+1101+J)
5035 POKE 1101+I*32+J,96
5040 NEXT J:NEXT I
5050 RETURN
```

Typing Hints . . .

You get two different colors on the screen by entering the direction names in reverse video. These appear in the listing as lower-case letters. To turn on the reverse video, just press the **SHIFT** and **0** keys. To turn it off, press the **SHIFT** and **0** keys again.

Highlights . . .

This program is longer than the other up-down program, but it is mostly the same. Only the subroutines are larger.

Each subroutine now has two loops. The outer (I) loop sets the number of times the arrows will move across the screen and controls the pitch of the SOUND command. The inner (J) loop controls the location of the arrow on the screen.

Each time an arrow is drawn, it is erased again by a POKE command to its location on the screen. This gives the appearance of movement.

Variables . . .

COUNT	Counter to set number of times arrow moves across the screen.
H	Array—stores arrow's screen locations.
I, J	Pointers used in array H.
P	Screen memory locations relevant to I and J.

Other variables are the same as in the last program.

Do-It-Yourself . . .

The program lets children enter directions only by pressing an arrow button. It would be nice if the program would also accept direction names: LEFT, RIGHT, UP, and DOWN.

37

HOW FAST ARE YOU?

For Parents and Teachers ...

This game helps children develop hand-eye coordination and concentration.

The computer draws a line from left to right across the screen. As soon as the child sees the line appear on the lefthand side of the screen, he or she presses the SPACE bar. When the SPACE bar or any other key is pressed, the line stops. The faster the child reacts, the better the score.

For Kids ...

How fast are you?

To find out, step right up and have a duel with the computer.

The computer will print a yellow block on the TV screen. It will tell you "GET READY!" Then a little red line will shoot across the screen, from left to right. As soon as you see the line appear on the left side of the screen, press the **SPACE** bar. The line will stop, and the computer will tell you how fast you are.

The slower you are, the longer the line on the screen. The faster you are, the shorter the line.

The computer will also tell you how much time passed between the time when the line first appeared and when you pressed the **SPACE** bar and stopped the line.

Keep playing the game to see if you can improve your time.

If the game gets too simple, change the rules. If you are right-handed, start pressing the **SPACE** bar with your left hand. If you are left-handed, start using your right hand. Or use your nose. Or your elbow. Or turn around and use a mirror.

Let your whole family try the game. Have a family championship. Maybe you can even teach your dog or cat how to play!

The Game ...

Program Name: **FAST**

```
50 REM *** HOW FAST ARE YOU
60 CLS
65 PRINT@34,"***  HOW FAST ARE YOU?  ***
"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 CLEAR 300
72 DIM M$(15)
75 FOR L=0 TO 15:READ M$(L):NEXT L
90 B$=" "
100 A$=""
105 CLS0
110 FOR L=0 TO 15:PRINT@L*32,CHR$(159);:
PRINT@L*32+30,CHR$(159);:NEXT L
120 FOR L=0 TO 30:PRINT@L,CHR$(159);:PRI
NT@L+480,CHR$(159);:NEXT L
140 PRINT@69,"***  GET READY!!  ***";
200 FOR PAUSE=1 TO 1000:NEXT PAUSE
```

```
205 X=0
210 A$=INKEY$:IF A$="" AND X < 60 THEN X
=X+1:SET(X,7,4):GOTO 210
231 IF X=0 THEN PRINT@164,"TOO SOON!   T
RY AGAIN!";:GOTO 250
232 A$=M$(INT(X/4))
235 GOSUB 3010:REM * CENTER LINE
240 PRINT@175-INT((LEN(A$))/2),A$;
250 FOR PAUSE=1 TO 2000:NEXT PAUSE
255 CLS
260 GOTO 100
3000 REM *** ADD TIME TO MESSAGE
3010 A$=A$+" ("
3020 A$=A$+ STR$(X)
3030 A$=A$+" COUNTS)"
3040 RETURN
5000 DATA ACE!!,PRO!,VERY FAST!,FAST
5004 DATA QUICK,AVERAGE,OKAY,SLOW,VERY S
LOW
5009 DATA SLOW AS A TURTLE,SLOW AS A SNA
IL,SLOW AS A SLOTH,WAKE UP!
5010 DATA YOU NEED HELP!
5012 DATA FORGET IT!
5014 DATA IS ANYBODY THERE?
```

Highlights ...

This game uses graphics. The commands on lines 110 and 120 draw the yellow playing field on the TV screen. The PRINT@ command draws the edges of the field.

The PRINT@ command on line 140 centers the text-area line for the "GET READY" message.

The SET command on line 210 draws the computer's line that sprints across the screen. The computer checks for keyboard input before plotting each point on the line.

If you press the **SPACE** bar before the computer starts drawing the sprint line, it tells you (on line 231): "TOO SOON! TRY AGAIN!" Then it starts all over.

The current column of the computer's sprinting line is represented by X. On line 232 the computer uses X (divided by

4) to point it to the right message to display on the screen. Line 240 centers the message and then prints it. The subroutine on line 3000 adds the child's time onto the message.

The STR$ function on line 3020 converts the value of X/4 into a string, so that it can be added to the message from the computer. X/4 represents the time in counts that it took for you to press the SPACE bar.

Variables ...

PAUSE	Delay loop counter.
A$	Message—computer tells you how well you did.
X	Current column position of the computer's sprint line.
M$	Array—holds messages.

Do-It-Yourself ...

The program always starts the race after it counts to 1000 on line 200. This makes it possible to anticipate the computer and get a better score than would otherwise be possible. To change this, you can put in an RND function and have the computer start the race randomly and unpredictably.

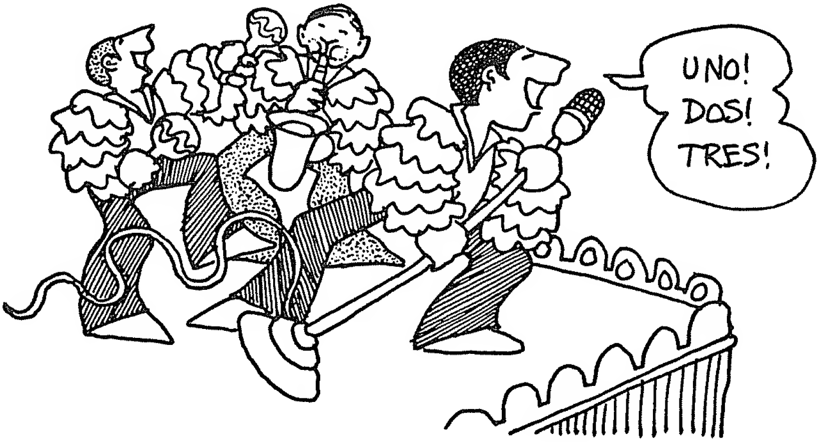
Also, it might make the race more interesting if you plotted a little figure (a horse, person, hare, or tortoise) to race across the track. You can move the figure by plotting it and erasing it at different positions across the screen.

You can also make the game more challenging by randomly starting the sprinting line (or figure) on the right or left side of the screen.

FOREIGN LANGUAGE

38

UNO! DOS! TRES!



For Parents and Teachers ...

This game helps children learn how to count from zero to 10 in Spanish.

For Kids ...

When you RUN this program, the screen turns blank. At the bottom, the computer asks "NUMBER (0-10)?" You can type any number from 0 to 10. Let's say you type a **5**. The computer puts a 5 on the screen and underneath it the word for 5 in Spanish:

```
5
CINCO
```

When the computer prints the number, it plays a special musical tone. The computer asks you to say the name of the number, first in English and then in Spanish.

If you keep practicing the numbers in Spanish you will soon be able to say them all without any help from the computer:
"CERO! UNO! DOS! TRES! CUATRO! CINCO! SEIS! SIETE! OCHO! NUEVE! DIEZ!"

The Game . . .

Program Name: **SPANISH**

```
 50 REM *** COUNT IN SPANISH
 60 CLS
 65 PRINT@33,"*** COUNT TO 10 IN SPANISH
 ***"
 67 FOR PAUSE=1 TO 1000:NEXT PAUSE
 76 B$="  "
 80 DIM M(10):DIM N$(10)
 82 FOR I=0 TO 10
 84 READ M(I)
 88 NEXT I
 90 FOR I=0 TO 10
 92 READ N$(I)
 94 NEXT I
 100 CLS
 110 FOR I=1 TO 11
 120 PRINT@329,B$:PRINT@329,"NUMBER (0-10
 )";
 122 INPUT A$
 130 IF VAL(A$)<0 OR VAL(A$)>10 THEN 120
 132 IF ASC(A$) <48 OR ASC(A$) >57 THEN 1
 20
 135 K=VAL(A$)
»165 PRINT@354,"SAY IN english AND spanis
 h"
 170 PRINT@398,K
 175 GOSUB 3010:REM * CENTER NUMBER
 190 SOUND M(K),6
 200 FOR PAUSE=1 TO 800:NEXT PAUSE
 220 PRINT@320,B$
```

```
230 PRINT:PRINT:PRINT
240 NEXT I
250 CLS
255 PRINT@397,"AGAIN";:INPUT A$
257 IF A$="Y" THEN 100
260 IF A$<>"N" THEN 250
270 CLS:END
3000 REM *** CENTER NUMBER SUBROUTINE
3010 N$=N$(K)
3020 PRINT@431-INT((LEN(N$))/2),N$
3030 RETURN
5000 DATA 89,108,125,133,147,159,170,176
,185,193,197
5010 DATA CERO,UNO,DOS,TRES,CUATRO,CINCO
,SEIS,SIETE,OCHO,NUEVE,DIEZ
```

Typing Hints ...

Line 165 contains the words English and Spanish in lower case. This means they are to be typed in reverse video. To get the reverse video, you need to press the **SHIFT** and **0** keys. To turn off the reverse video, press the **SHIFT** and **0** keys again.

Highlights ...

The musical tones that accompany each number are in the DATA command on line 5000. The Spanish names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the M array on lines 82 to 88. The Spanish numbers are read into the N$ array on lines 90 to 94.

The program lets you choose any 11 numbers from 0 to 10. You can count from 0 to 10. Or you can repeat the number 5 (CINCO) 11 times. The major program loop begins on line 110 and ends on line 240.

The program INPUTs your chosen numeric digits into the string variable A$. The program moves the value of A$ into the variable K (line 135) and fetches the Spanish name for the number you have chosen.

The Spanish names are different lengths. The subroutine beginning on line 3000 centers them (roughly) under the number.

A trap is used on lines 130 and 132. If you type something other than a number in response to the question on line 120, the trap takes over. It causes the computer to jump to line 120, initialize the screen, and ask you the same question over again. When the computer does this, it uses the trap again. This readies the computer for any new errors.

Variables ...

PAUSE	Delay loop counter.
B$	Blank—erases old numbers and names.
M	Array—stores musical tones for each Spanish number name.
I	Loop counter.
A$	Stores your answer to "PLAY AGAIN?"
K	The numeric value of the number you choose.
N$	Array—stores Spanish words for each number from 0 to 10.

Do-It-Yourself ...

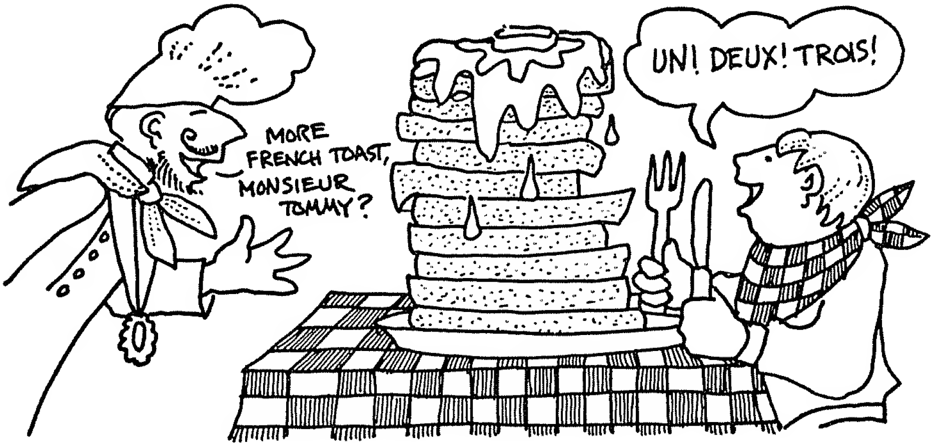
The program only accepts numbers (0, 1, 2, etc.) to indicate which Spanish number name you want to practice. You might modify the program to accept English number names—"zero," "one," "two," and so on. Or, you could have the program accept Spanish number names—"uno," "dos," "tres," etc. This would help the children learn how to spell the names.

A neat feature for the program to have would be automatic counting. At the beginning and end of the game, the program could count from 0 to 10 in Spanish. It would display all the numbers and play all the musical tones.

Also, the program could be expanded to Spanish words and Spanish phrases. It could also display the phonetic pronunciation of each of the numbers, words, and phrases to help the children say the words correctly.

39

UN! DEUX! TROIS!



For Parents and Teachers ...

This game helps children learn how to count from zero to 10 in French.

For Kids ...

When you RUN this program, the screen turns blank. At the bottom, the computer asks "NUMBER (0-10)?" You can type any number from 0 to 10. Let's say you type a **5**. The computer puts a 5 on the screen and underneath it the word for 5 in French:

```
        5
      CINQ
```

When the computer prints the number, it plays a special musical tone. The computer asks you to say the name of the number, first in English and then in French.

If you keep practicing the numbers in French you will soon be able to say them all without any help from the computer: "NUL! UN! DEUX! TROIS! QUATRE! CINQ! SIX! SEPT! HUIT! NEUF! DIX!"

The Game ...

Program Name: **FRENCH**

```
50 REM *** COUNT IN FRENCH
60 CLS
65 PRINT@34,"*** COUNT TO 10 IN FRENCH *
**"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
76 B$=" "
80 DIM M(10):DIM N$(10)
82 FOR I=0 TO 10
84 READ M(I)
88 NEXT I
90 FOR I=0 TO 10
92 READ N$(I)
94 NEXT I
100 CLS
110 FOR I=1 TO 11
120 PRINT@329,B$:PRINT@329,"NUMBER (0-10
)";
122 INPUT A$
130 IF VAL(A$)<0 OR VAL(A$)>10 THEN 120
132 IF ASC(A$) <48 OR ASC(A$) >57 THEN 1
20
135 K=VAL(A$)
165 PRINT@354,"SAY IN english AND french
"
170 PRINT@398,K
175 GOSUB 3010:REM * CENTER NUMBER
190 SOUND M(K),6
200 FOR PAUSE=1 TO 800:NEXT PAUSE
220 PRINT@320,B$
230 PRINT:PRINT:PRINT
```

```
240 NEXT I
250 CLS
255 PRINT@397,"AGAIN";:INPUT A$
257 IF A$="Y" THEN 100
260 IF A$<>"N" THEN 250
270 CLS:END
3000 REM *** CENTER NUMBER SUBROUTINE
3010 N$=N$(K)
3020 PRINT@431-INT((LEN(N$))/2),N$
3030 RETURN
5000 DATA 89,108,125,133,147,159,170,176
,185,193,197
5010 DATA NUL,UN,DEUX,TROIS,QUATRE,CINQ,
SIX,SEPT,HUIT,NEUF,DIX
```

Typing Hints ...

Line 165 contains the words English and French in lower case. This means they are to be typed in reverse video. To get the reverse video, you need to press the **SHIFT** and **0** keys. To turn off the reverse video, press the **SHIFT** and **0** keys again.

Highlights ...

The musical tones that accompany each number are in the DATA command on line 5000. The French names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the M array on lines 82 to 88. The French numbers are read into the N$ array on lines 90 to 94.

The program lets you choose any 11 numbers from 0 to 10. You can count from 0 to 10. Or you can repeat the number 5 (CINQ) 11 times. The major program loop begins on line 110 and ends on line 240.

The program INPUTs your chosen numeric digits into the string variable A$. The program moves the value of A$ into the variable K (line 135) and fetches the French name for the number you have chosen.

The French names are different lengths. The subroutine beginning on line 3000 centers them (roughly) under the number.

A trap is used on lines 130 and 132. If you type something other than a number in response to the question on line 120, the trap takes over. It causes the computer to jump to line 120, initialize the screen, and ask you the same question over again. When the computer does this, it uses the trap again. This readies the computer for any new errors.

Variables . . .

PAUSE	Delay loop counter.
B$	Blank—erases old numbers and names.
M	Array—stores musical tones for each French number name.
I	Loop counter.
A$	Stores your answer to "PLAY AGAIN?"
K	The numeric value of the number you choose.
N$	Array—stores French words for each number from 0 to 10.

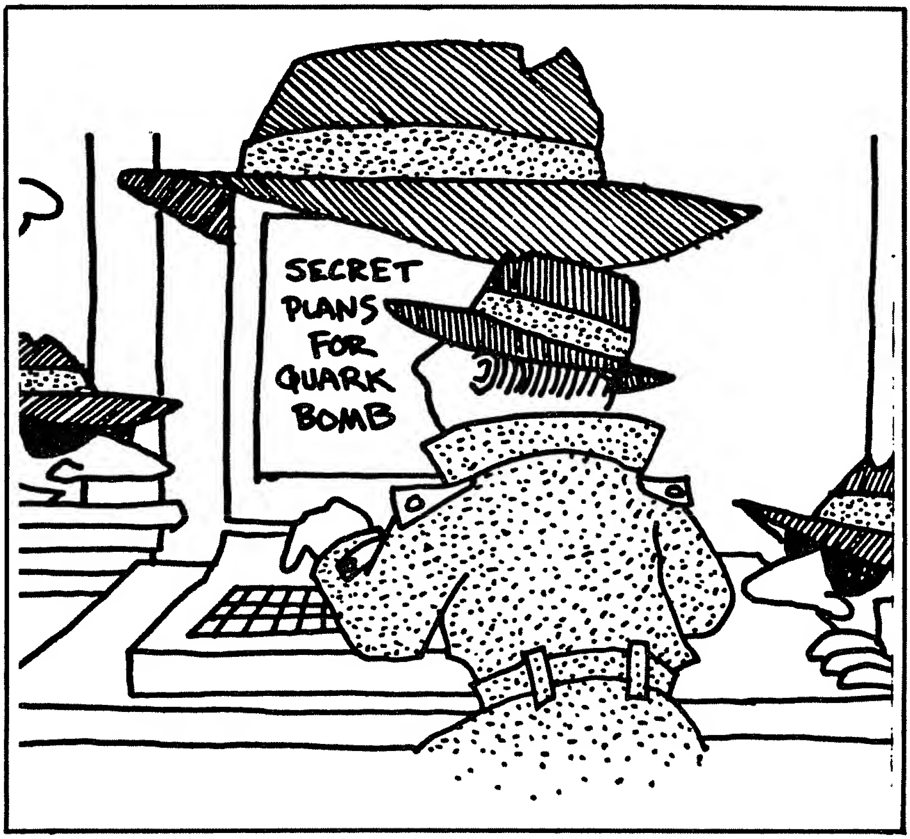
Do-It-Yourself . . .

The program only accepts numbers (0, 1, 2, etc.) to indicate which French number name you want to practice. You might modify the program to accept English number names—"zero," "one," "two," and so on. Or, you can have the program display only the English name and the digit. Then the child has to type in the correct French name—"un," "deux," "trois," etc.

It would also be neat if the program could automatically count and display the numbers. At the beginning and end of the game, the program would count from zero to 10 in French. It would display all the numbers and play all the musical tones.

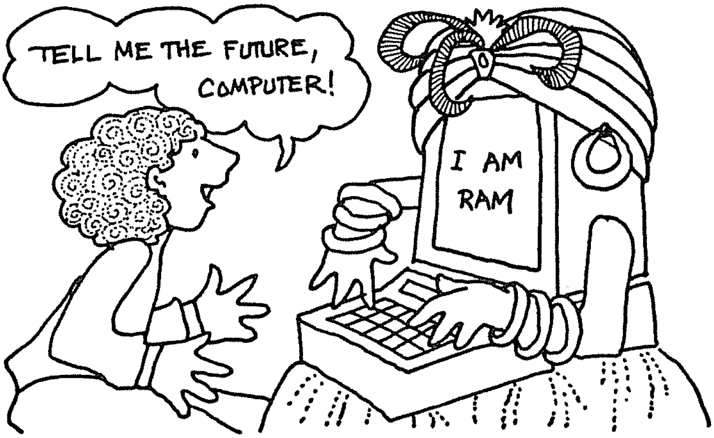
Also, the program could be expanded to French words and French phrases. It could also display the phonetic pronunciation of each of the numbers, words, and phrases to help the children say the words correctly.

IMAGINATION



40

THE FORTUNE TELLER



For Parents and Teachers ...

This is an imagination game. The computer plays the part of a fortune teller and pretends it can see into the child's future. This program will motivate children to develop their writing and typing skills.

For Kids ...

Imagine that it's late at night and you are asleep.

A crash of thunder wakes you. You hear eerie music coming up the stairs. You sneak downstairs to see what is making the music.

It's the computer. Strange messages appear on the computer's TV screen:

```
I AM RAM. I AM A WIZARD.

I AM 10,000 YEARS OLD.

I LIVE INSIDE YOUR COMPUTER.
```

This isn't real. You pinch yourself to see if you are dreaming. Ouch! Nope.

New messages appear:

```
WHEN YOU TURN ON YOUR COMPUTER,
I WAKE UP.

THEN I CAN SEE THE FUTURE.

I CAN SEE YOUR FUTURE!
```

All of a sudden, you start to shiver.

The messages continue:

```
ASK ME ANY QUESTION
WITH A YES OR NO ANSWER
UP TO 79 CHARACTERS LONG.

WHAT IS YOUR QUESTION?
```

Wow! You could use this computer wizard to predict the outcome of elections, or bet on horse races and make a million dollars. Or you could become a superstar reporter and find out the news even before it happens!

What should be your first question? You have it! You type madly.

The wizard flashes a new message:

```
I AM GAZING AT MY CRYSTAL BALL ...

                GAZING ...

                GAZING ...

                GAZING ...
```

I SEE YOUR FUTURE!

YOUR QUESTION:

WILL I EVER SCORE A MILLION POINTS
IN THE NEW ARCADE GAME
MEATBALL WARS?

THE ANSWER IS ...

The wizard pauses for dramatic suspense.

You press your nose against the glass of the TV screen. You are so close the words fuzz up. Your eyes start to cross. "What's the answer?" you cry.

The wizard finally answers:

YES!!

"Hurray!" you cheer.

You run off to tell the other members of your family about the fortune teller who lives inside your computer.

The Game ...

Program Name: **FORTUNE**

```
50 REM *** FORTUNE TELLER
60 CLS
65 PRINT@34,"***  THE FORTUNE TELLER  **
*"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 CLS
102 PRINT:PRINT:PRINT:PRINT:PRINT
110 PRINT"I AM RAM.  I AM WIZARD."
120 PRINT
130 PRINT"I AM 10,000 YEARS OLD."
140 PRINT
150 PRINT"I LIVE INSIDE YOUR COMPUTER."
160 FOR PAUSE=1 TO 3000:NEXT PAUSE
170 CLS
172 PRINT:PRINT:PRINT:PRINT
```

```
180 PRINT"WHEN YOU TURN ON YOUR COMPUTER
,"
185 PRINT
190 PRINT"I WAKE UP."
200 PRINT
210 PRINT"THEN I CAN SEE THE FUTURE."
220 PRINT
230 PRINT"I CAN SEE your FUTURE!"
235 PRINT
240 FOR PAUSE=1 TO 3000:NEXT PAUSE
250 CLS
255 M$=""
257 PRINT:PRINT:PRINT:PRINT
260 PRINT"ASK ME ANY QUESTION WITH A"
265 PRINT
270 PRINT" yes OR no ANSWER."
280 PRINT
290 PRINT"WHAT IS YOUR QUESTION":INPUT M
$
300 CLS
304 PRINT"I AM GAZING AT MY CRYSTAL BALL
...................."
305 PRINT
306 PRINT
307 FOR PAUSE=1 TO 100:NEXT PAUSE
310 FOR I=1 TO 3
320 PRINT"     gazing..."
325 FOR PAUSE=1 TO 400:NEXT PAUSE
330 PRINT
340 NEXT I
345 PRINT
347 CLS
350 PRINT"I SEE YOUR FUTURE!"
360 PRINT
370 FOR PAUSE=1 TO 800:NEXT PAUSE
372 PRINT
373 PRINT"YOUR QUESTION:"
374 PRINT:PRINTM$
375 PRINT:PRINT
380 PRINT"THE ANSWER IS...";
390 FOR PAUSE=1 TO 2000:NEXT PAUSE
400 IF RND(2)=1 THENPRINT"yes!!":GOTO 43
0
410 PRINT"no!!"
```

```
430 FOR PAUSE=1 TO 1000:NEXT PAUSE
440 CLS
450 PRINT"WANT ME TO TELL YOUR"
460 PRINT"FUTURE AGAIN";:INPUT A$
470 IF A$="Y" THEN 250
480 IF A$<>"N" THEN 440
490 CLS:END
```

Typing Hints ...

Some lines contain words in lower case. This means they are to be typed in reverse video. You can get the letters to appear in reverse video on lines 230, 270, 320, 400, and 410 by pressing the SHIFT and 0 keys. To turn the reverse video off, press the SHIFT and 0 keys again.

Highlights ...

This program consists almost entirely of PRINT statements.

The key to the program is on line 400. There the computer swami "flips a coin" to decide your future. The RND function makes the computer choose either a 1 or a 2. If the computer chooses a 1, it answers your question "YES!!" If the computer chooses a 2, it answers your question "NO!!"

What makes this program a success, of course, is obviously not programming knowhow. Instead it is atmosphere and imagination. Sit down and try asking the computer to predict your future. Before you know it, you will be asking the computer some pretty serious questions. It's easier than you think to come under the fortune teller's spell.

Variables ...

PAUSE	Delay loop counter.
A$	Accepts your answer to "WANT ME TO TELL YOUR FUTURE AGAIN?"

M$	Accepts child's question.
I	Loop counter—prints "GAZING . . ." three times.

Do-It-Yourself . . .

You can add all sorts of bells and whistles to this program to heighten the illusion that a real 10,000-year-old wizard lives inside the computer. For example, the wizard can ask the child his or her name. Information in DATA statements can correlate with a particular name and the wizard can impress the child with how much he knows about him or her.

You can also add SOUND commands that make sound effects and eerie noises. You might even consider adding a crystal ball on the screen or a glimpse of the wizard's ancient face.

This program is an example of how a good game can be 90% imagination and only 10% perspiration.

41

SECRET AGENT

For Parents and Teachers ...

This is an imagination game. It takes words and sentences and turns them into secret codes for children to pass around and try to figure out. This game might act as an incentive to encourage your children to practice writing and typing on the computer.

For Kids ...

Pretend you are agent Triple-Nine. You have a secret message that you have to deliver to the president of a small country nestled in the Andes Mountains in South America. The message contains the plans to a powerful, new, top-secret Quark bomb. The president needs this bomb to defend his country against an imminent attack of robot guerillas from a neighboring country.

You are almost ready to board your private jet and fly to the president's country. But first, in case you meet with foul play, you need to translate the bomb plans into a secret code. Then, even if the plans fall into the enemy's hands, they will be useless.

You can invent a secret code. Then you can take each letter of each word in the bomb plans and translate it.

But this would take forever. Anyway, you already have a coding machine. It's your computer.

You turn on the computer, load the Secret-Agent program, and type **RUN**. The computer asks you for your code name. You type **999**. "GOOD NAME!" says the computer. "I LIKE IT!"

You type in the secret plans, one line at a time. A message flashes on the TV screen: "CODING MACHINE NOW WORKING." Moments later the coded bomb plans appear on the screen. You copy them down and destroy the original plans. You board your airplane and head for South America.

The Game . . .

Program Name: **SECRET**

```
 50 REM *** SECRET AGENT
 60 CLS
 65 PRINT@34,"***  SECRET AGENT GAME  ***
"
 66 FOR PAUSE=1 TO 1000:NEXT PAUSE
 100 CLS
 110 PRINT@129,"WHAT IS YOUR CODE NAME?"
 112 INPUT N$
 120 CLS
 130 PRINT@129,"GOOD NAME!  I LIKE IT!"
 140 PRINT
 150 PRINTN$;", WHAT IS YOUR SECRET"
 155 PRINT"MESSAGE";:INPUT M$
 160 CLS:M2$=""
➧170 PRINT@35,"coding machine working"
 180 FOR PAUSE=1 TO 3000:NEXT PAUSE
 190 FOR I=1 TO LEN(M$)
 195 IF MID$(M$,I,1)=" "THEN M2$=M2$+" ":
```

```
GOTO 210
200 T$=MID$(M$,I,1)
202 T=ASC(T$)
204 T=T+1
206 T$=CHR$(T)
208 M2$=M2$+T$
210 NEXT I
220 CLS
230 PRINT@129,"ORIGINAL MESSAGE:"
240 PRINT
250 PRINTM$
260 PRINT:PRINT
270 PRINT" CODED MESSAGE:"
280 PRINT
290 PRINTM2$
300 PRINT:PRINT:PRINT
310 PRINT"PLAY AGAIN <Y/N>";:INPUT A$
320 IF A$="Y" THEN CLS:GOTO 150
330 IF A$<>"N" THEN 310
340 CLS:END
1800 FOR PAUSE=1 TO 3000:NEXT PAUSE
```

Typing Hints ...

The message on line 170 is in lower case. This means it is to be typed in reverse video. You can make the reverse video message by pressing the **SHIFT** and **0** keys. To turn the reverse video off, press the **SHIFT** and **0** keys again.

Highlights ...

This is an extremely simple program. It begins and ends with some PRINT commands. In between is a small loop on lines 190 to 210 that translates your message from English to secret code.

Line 200 is the key line. On line 200 the computer takes a letter or number in your message, converts it to its ASCII code value, then adds one. Next, it converts this number back into some new letter, character, or punctuation symbol. The message in its original form is stored in M$. The new, coded message is stored in M2$.

Variables ...

PAUSE	Delay loop counter.
A$	Your answer to question "NEW MESSAGE?"
N$	Your secret-agent name.
M$	Original message.
M2$	Coded message.
I	Counter for coding loop.

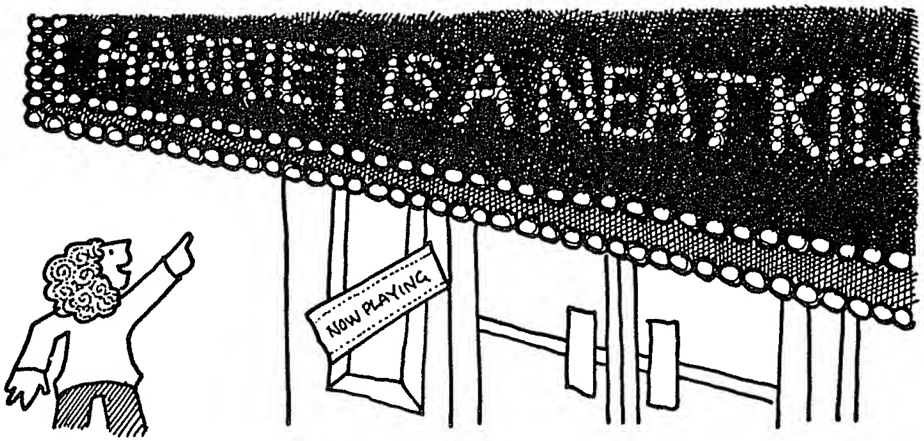
Do-It-Yourself ...

This game is good at coding secret messages but not at decoding them. You can modify the program to make it into a new, secret-message decoder. To do that you need to change line 200. Instead of adding 1 to the ASCII value of MID$(M$,I,1), you have the program subtract 1.

You might also modify the program so it stores your child's secret messages on tape or disk. That way the messages don't have to be copied by hand back into the computer each time the child wants to code or decode them.

42

NEWS BULLETIN!



For Parents and Teachers ...

This is an imagination game. Children pretend the computer is a giant marquee in a big city. They get to think up any message they want to put on that marquee.

This game will encourage children to compose unusual sentences and messages. It will help them develop their writing and typing skills.

For Kids ...

How would you like to see your name on TV? Or in the movies? How would it feel to go to a theater and see your name up there in lights along with famous movie stars? Wouldn't it be great to be mentioned on a "World News" program some night?

Or, have you ever thought about making up your own newspaper headlines? They could be serious, or they could be crazy, weird, or funny.

You can make up your own headlines. And you can be in them, too! How? On your computer—using the News Bulletin game.

When you load and run the News Bulletin game, it asks you to enter your news bulletin. Type in whatever you think up. The game turns your message into headline letters that march across the TV screen. Each letter makes its own special musical note.

Now turn out the lights in your room. Sit down on the floor beneath the glowing computer TV screen. Pretend it is nighttime and you are in a huge city. Look up. Your bulletin appears, and millions of people can see it!

The Game . . .

Program Name: **BULLETIN**

```
50 REM *** NEWS BULLETIN
60 CLS
65 PRINT@37,"*** NEWS BULLETIN! ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 CLEAR 500
80 DIM MUSIC(90)
90 LOW=48:HIGH=57:GOSUB 3010
92 LOW=65:HIGH=90:GOSUB 3010
93 READ MUSIC(32)
100 CLS
110 PRINT@224,"YOUR NEWS BULLETIN:":INPU
T M$
115 IF M$="" THEN CLS:END
120 CLS:C=0
130 PRINT@2,"*** YOUR NEWS BULLETIN!! **
*"
135 P=18
137 PRINT@0," "
138 PRINT@8,"*** BULLETIN ***"
140 Y=2
142 P=18
```

```
144 IF LEN(M$)<=C+18 THEN 150
146 IF MID$(M$,C+18,1)<>" " THEN GOSUB 3
510
150 X=1
160 C=C+1
165 IF C<=LEN(M$) THEN 170
167 FOR PAUSE=1 TO 1000:NEXT PAUSE
168 GOTO 120
170 PRINT@Y*32+X+7,MID$(M$,C,1);
171 T=ASC(MID$(M$,C,1))
172 IF T=32 THEN 180
173 IF T>=48 AND T<=57 THEN 180
174 IF T>=65 AND T<=90 THEN 180
175 SOUND MUSIC(32),8
176 GOTO 190
180 SOUND MUSIC(ASC(MID$(M$,C,1))),4
190 FOR PAUSE=1 TO 60:NEXT PAUSE
210 X=X+1:IF X<=P THEN 160
250 Y=Y+1:IF Y<=11 THEN 142
260 CLS:GOTO 135
3000 REM *** LOAD MUSICAL TONES
3010 FOR I= LOW TO HIGH
3020 READ M
3030 MUSIC(I)=M
3040 NEXT I
3050 RETURN
3500 REM *** SHRINK LINE SUBROUTINE
3510 P=17
3520 IF MID$(M$,C+P,1)=" " THEN RETURN
3530 P=P-1:IF P=>5 THEN 3520
3550 RETURN
5000 DATA 89,99,108,117,125,133,140,147,
153,159
5010 DATA 165,170,176,180,185,189,193,19
7,200,204
5020 DATA 207,210,213,216,218,221,223,22
5,227,229
5030 DATA 231,232,234,236,237,238,239
```

Highlights ...

The Bulletin game has a musical tone associated with each letter in the alphabet and with the numbers 0 through 9. These tones are loaded into the MUSIC array by the subroutine beginning on line 3000. This subroutine is called right at the beginning of the program on lines 90 and 92.

The computer looks ahead to the end of each new line. If a word is too big to fit on the line, the computer bumps that word to the next line. On the current line it prints only the words preceding that word in the bulletin. (See the subroutine beginning on line 3500, and see lines 144 and 146.) Each line, when printed, will be 18 letters long; up to 10 lines will fit on your screen.

The computer has a musical tone associated with each letter or number. If the computer encounters a space or punctuation symbol, it prints it on the screen, then plays a special tone.

Variables . . .

PAUSE	Delay loop counter.
M$	Your bulletin.
MUSIC	Array—stores the musical tones associated with each letter and number.
LOW	Low value in MUSIC where tones are stored.
HIGH	High value in MUSIC where tones are stored.
M	A single musical tone read from DATA statement.
C	Points to current letter or number in bulletin string (M$) that is about to be printed.
X	Column position for current letter or number.
Y	Row position for current line.
P	Final column position for letter or number on the current line on the TV screen.

T	The ASCII value of the current letter or number in the bulletin.
I	Loop counter for reading tones into MUSIC.

Do-It-Yourself . . .

Once you start the News Bulletin program, it keeps going. This is so you won't have to tend to it. You can leave a message for a member of your family and leave town. The bulletin program will display that message even if your family member arrives hours later.

You can change the program so that it stops after it repeats the bulletin a certain number of times. You can have it jump back to line 100. Then, if you want to exit the program, you just press the **ENTER** button when the computer asks you for a new bulletin. (See line 115.)

You might also change the way the letters march across the TV screen. Real electronic banners set whole words in motion. The words sail across the giant screen from right to left, like a fleet of ships in a regatta. You can do that on your computer. You need to pull letters and words from M$ and start them out on the right side of the screen. Then march them all together over to the left side of the screen. When all the words have marched across the screen, go back to the beginning of M$ and start again.

APPENDIX

PITCH VALUES FOR MUSICAL NOTES

F	5	F	197
F$^{\sharp}$, G$^{\flat}$	19	F$^{\sharp}$, G$^{\flat}$	200
G	32	G	204
G$^{\sharp}$, A$^{\flat}$	45	G$^{\sharp}$, A$^{\flat}$	207
A	58	A	210
A$^{\sharp}$, B$^{\flat}$	69	A$^{\sharp}$, B$^{\flat}$	213
B	78	B	216
Middle C	89	C	218
C$^{\sharp}$, D$^{\flat}$	99	C$^{\sharp}$, D$^{\flat}$	221
D	108	D	223
D$^{\sharp}$, E$^{\flat}$	117	D$^{\sharp}$, E$^{\flat}$	225
E	125	E	227
F	133	F	229
F$^{\sharp}$, G$^{\flat}$	140	F$^{\sharp}$, G$^{\flat}$	231
G	147	G	232
G$^{\sharp}$, A$^{\flat}$	153	G$^{\sharp}$, A$^{\flat}$	234
A	159	A	236
A$^{\sharp}$, B$^{\flat}$	165	A$^{\sharp}$, B$^{\flat}$	237
B	170	B	238
C	176	C	239
C$^{\sharp}$, D$^{\flat}$	180	C$^{\sharp}$, D$^{\flat}$	241
D	185	D	242
D$^{\sharp}$, E$^{\flat}$	189	D$^{\sharp}$, E$^{\flat}$	243
E	193	E	244

Radio Shack®